Mechatronics

Fundamentals and Applications

VISIT…

LANZAROTE
Caliente.COM

Mechatronics

Fundamentals and Applications

Edited by

Clarence W. de Silva
The University of British Columbia
Vancouver, Canada

Farbod Khoshnoud
Brunel University London
Uxbridge, UK

Maoqing Li
Xiamen University
China

Saman K. Halgamuge
University of Melbourne
Melbourne, Australia

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

MATLAB® and Simulink® are trademarks of The MathWorks, Inc. and are used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® and Simulink® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® and Simulink® software.

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2016 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20150922

International Standard Book Number-13: 978-1-4822-3932-4 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface ... vii
Acknowledgments ... ix
Editors ... xi
Contributors ... xiii

1. **Mechatronic Engineering** ... 1
   Clarence W. de Silva

Section I Fundamentals

2. **Modeling for Control of Rigid Bodies in 3-D Space** ... 17
   Ibrahim Esat, Minoo Dabestani, William Mortel, and Steve Sewell

3. **Mechanics of Materials** ... 37
   Yong Kang Chen

4. **Control of Mechatronic Systems** ... 85
   Kok Kiong Tan and Andi Sudjana Putra

5. **Introduction to Sensors and Signal Processing** ... 125
   Diogo Montalvão

6. **Bio-MEMS Sensors and Actuators** ... 221
   Farbod Khoshnoud, Clarence W. de Silva, and Ibrahim Esat

7. **System Identification in Human Adaptive Mechatronics** ... 253
   Bin Xi and Clarence W. de Silva

8. **Intelligent Robotic Systems** ... 295
   Muhammad Tahir Khan, Clarence W. de Silva, and Javaid Iqbal

Section II Applications

9. **Automated Mechatronic Design Tool** ... 323
   Saeed Behbahani, Leila Pezeshki, and Clarence W. de Silva

10. **Design Evolution of Mechatronic Systems** ... 359
    Lalith B. Gamage

11. **Mechatronic Design of Unmanned Aircraft Systems** ... 403
    Feng Lin, Fei Wang, Xiangxu Dong, Kemao Peng, and Ben M. Chen

12. Self-Powered and Bio-Inspired Dynamic Systems 453
Farbod Khoshnoud and David J. Dell

13. Visual Servo Systems for Mobile Robots 499
Haoxiang Lang and Clarence W. de Silva

14. Robotic Learning and Applications 529
Ying Wang and Bashan Zuo

15. Neuromechatronics with *In Vitro* Microelectrode Arrays 567
Dulini Mendis, Steven Petrou, and Saman K. Halgamuge

Index 589

Preface

With individual chapters authored by professionals in their respective topics, this book provides a convenient and up-to-date reference with information on the latest developments of mechatronics for engineers, designers, researchers, educators, and students. The presented material includes methodologies that encompass modeling, analysis, design, control, monitoring, and diagnosis of mechatronic systems and state-of-the-art mechatronic systems and technologies. The book consists of 15 chapters, grouped into two sections: fundamentals and applications. Cross-referencing is used when necessary to indicate other places in the book where further information on a particular topic is provided.

In the book, equal emphasis is given to theory and practical application. The chapters cover fundamentals and applications of mechatronic devices and systems with specific treatment of related topics, including modeling and analytical formulations, simulation methods, design approaches, control techniques, practical tools, and cutting-edge systems and applications, illustrated using examples and case studies.

Mechatronics concerns synergistic and concurrent use of mechanics, electronics, computer engineering, and intelligent control systems in modeling, analyzing, designing, developing, and implementing smart electromechanical products. As modern machinery and electromechanical devices are typically being controlled using analog and digital electronics and computers, the technologies of mechanical engineering in such systems can no longer be isolated from those of electronic and computer engineering. For example, in a robot system or a micro-machine, mechanical components are integrated with analog and digital electronic components to provide single functional units or products. Similarly, devices with embedded and integrated sensing, actuation, signal processing, and control have many practical advantages. In the framework of mechatronics, a unified approach is taken to integrate different types of components and functions, both mechanical and electrical, in modeling, analysis, design, and implementation with the objective of harmonious operation that meets a desired set of performance specifications.

In the mechatronic approach, a multidomain (mixed) system, consisting of subsystems that have a primarily mechanical (including fluid and thermal) or a primarily electrical (including electronic) character, is treated using integrated engineering concepts. In particular, electromechanical analogies, consistent energy transfer (e.g., kinetic, potential, thermal, fluid, electrostatic, and electromagnetic energies) through energy ports, and integrated design methodologies may be incorporated, resulting in benefits with regard to performance, efficiency, reliability, and cost.

Mechatronics has emerged as a bona fide field of practice, research, and development and simultaneously as an academic discipline in engineering. The present book is geared toward the focus on integrated research and practice as related to electromechanical and multidomain systems. In view of the analytical methods, practical considerations, design issues, and experimental techniques that are presented throughout the book, it serves as a useful reference tool and an extensive information source for engineers in industry and laboratories, researchers, and students in the field of mechatronics.

This book is an outgrowth of the Distinguished Visiting Fellowship of the Royal Academy of Engineering, UK, held by Clarence de Silva. The fellowship visit was organized by Farbod Khoshnoud. Through that fellowship, University of Hertfordshire and University of Oxford were visited, and among other activities, a workshop on mechatronics and applications was held. Many of the chapter authors of this book were speakers at that workshop.

Clarence W. de Silva
Farbod Khoshnoud
Maoqing Li
Saman K. Halgamuge

MATLAB® is a registered trademark of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA
Tel: 508-647-7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com

Acknowledgments

This book is an outgrowth of the Distinguished Visiting Fellowship of the Royal Academy of Engineering, UK, held by Clarence de Silva. The visit was organized by Farbod Khoshnoud. Through that fellowship, University of Hertfordshire and University of Oxford were visited, and among other activities, a workshop on mechatronics and applications was held. Many of the chapter authors of this book were speakers at that workshop. We gratefully acknowledge here the funding provided by the Royal Academy of Engineering, UK, which facilitated preparation of the book manuscript. We wish to express our gratitude to the authors of the chapters for their valuable and highly professional contributions. We are very grateful to Jonathan Plant, executive editor–engineering, Taylor & Francis, CRC Press, for his enthusiasm, encouragement, and support throughout the project. Editorial and production staff at CRC Press and its affiliates, particularly Iris Fahrer, project editor, and Adel Rosario, project manager, have done an excellent job in getting the book out in print. Finally, we wish to lovingly acknowledge the patience and understanding of our wives.

Editors

Clarence W. de Silva, PhD, is a professor of mechanical engineering and occupies the Senior Canada Research Chair Professorship in Mechatronics & Industrial Automation at the University of British Columbia (UBC), Vancouver, Canada. A professional engineer (P.Eng.), he is also a fellow of ASME, IEEE, the Canadian Academy of Engineering, and the Royal Society of Canada. He is a Peter Wall Scholar at UBC; a professorial fellow at University of Melbourne; distinguished visiting fellow of the Royal Academy of Engineering, UK; Lilly Fellow at Carnegie Mellon University; NASA-ASEE Fellow; Senior Fulbright Fellow at Cambridge University; fellow of the Advanced Systems Institute of BC; Killam Fellow; and Erskine Fellow at University of Canterbury. He has held the NSERC-BC Packers Senior Chair Professorship at UBC; Mobil Endowed Chair Professorship at the National University of Singapore; honorary chair professorship at the National Taiwan University of Science and Technology; and honorary professorship at Xiamen University, China. His awards include the Paynter Outstanding Investigator Award and the Takahashi Education Award of ASME; Killam Research Prize; and Outstanding Engineering Educator Award of IEEE Canada. Dr. de Silva has served as editor/associate editor of 14 journals, including *ASME* and *IEEE Transactions*; and is the editor-in-chief of the *International Journal of Control and Intelligent Systems*. He earned PhD degrees from the Massachusetts Institute of Technology, USA (1978) and the University of Cambridge, UK (1998) and an Honorary DEng. from the University of Waterloo (2008).

Farbod Khoshnoud, PhD, PGCE, CEng, MIMechE, FHEA, is a lecturer in the Department of Mechanical Engineering at Brunel University, London, UK. His current research areas include self-powered dynamic systems and biologically inspired dynamic systems. Dr. Khoshnoud was a lecturer at the University of Hertfordshire before joining Brunel. He was a visiting scientist and postdoctoral researcher in the industrial automation laboratory, Department of Mechanical Engineering, at the University of British Columbia (UBC), Vancouver, during 2007–2012, and was a visiting researcher at the California Institute of Technology during 2009–2011. He has carried out postdoctoral research in the Department of Civil Engineering at UBC during 2005–2007. Dr. Khoshnoud earned his PhD in mechanical engineering from Brunel University, London, in 2005. He has worked in industry as a mechanical engineer for more than six years. He is an associate editor of the *Journal of Control and Intelligent Systems*, ACTA Press.

Maoqing Li is professor of control science and engineering and system engineering in the School of Information Science and Technology, Xiamen University, China. He graduated from Xiamen University in 1978 and joined its faculty in the same year. He is the vice chairman of the Society on National Computer Education in Universities and Colleges, China, and editor of the *International Journal of Systems and Control*. He served as the president of the International Conference on Control and Automation in 2002 and 2010, chairman of the organizing committee of the 10th Conference on Manufacturing and Management, chairman of the organizing committee of the International Conference on Environment and Information in 2005, and chairman of the organizing committee of the first International Conference on Computer Science and Education in 2006. He has authored more than

200 technical papers and 6 monographs/textbooks. He received the first Fujian Province Youth Science and Technology Award.

Saman K. Halgamuge, PhD, is a professor in the Department of Mechanical Engineering and the school-wide initiative on biomedical engineering and associate dean (international) for the Melbourne School of Engineering, The University of Melbourne. He graduated with Dipl.-Ing and PhD degrees in electrical engineering from the Technical University of Darmstadt, Germany, and with a BSc in electronic and telecommunication engineering from the University of Moratuwa, Sri Lanka. His research interests are in big data analytics and optimization and in particular bioinspired methods focusing on applications in mechanical engineering and bioengineering. These applications vary from sensor networks in irrigation, smart grids, and sustainable energy generation to bioinformatics and neuro-engineering. Since arriving in Australia in 1996, Professor Halgamuge has obtained research grants totaling more than $8.3 million. These grants include funds worth $4.3 million from 17 Australian Research Council grants and a NHMRC project grant, local and European industry, contracts and grants from other research funding agencies and a large-scale ARC network, and infrastructure grants worth about $4 million. He has completed supervision of 27 PhD students and currently supervises a group of 13 PhD students. Professor Halgamuge is an associate editor of *BMC Bioinformatics* and founding co-editor of *Frontiers of Mechanical Engineering*, mechatronics section. He has published more than 250 research papers, including a research book, 5 edited books, 20 book chapters, 90 journal articles, and more than 130 refereed conference papers attracting 5200 Google Scholar (h-factor: 29) and 2300 ISI citations. He is listed among the top 1% most cited researchers by ISI's ESI. His full profile is found at http://scholar.google.com.au/citations?sortby=pubdate&hl=en&user=9cafqywAAAAJ&view_op=list_works.

Contributors

Saeed Behbahani
Department of Mechanical Engineering
Isfahan University of Technology
Isfahan, Iran

Ben M. Chen
Department of Electrical and Computer Engineering
National University of Singapore
Singapore

Yong Kang Chen
School of Engineering and Technology
University of Hertfordshire
Hatfield, Hertfordshire, United Kingdom

Minoo Dabestani
Dynamical Systems Technologies Ltd.
Cuffley, Hertfordshire, United Kingdom

Clarence W. de Silva
Department of Mechanical Engineering
The University of British Columbia
Vancouver, British Columbia, Canada

David J. Dell
School of Engineering and Technology
University of Hertfordshire
Hatfield, Hertfordshire, United Kingdom

Xiangxu Dong
Temasek Laboratories
National University of Singapore
Singapore

Ibrahim Esat
Department of Mechanical Engineering
Brunel University
Uxbridge, Middlesex, United Kingdom

Lalith B. Gamage
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

Saman K. Halgamuge
Department of Mechanical Engineering
University of Melbourne
Parkville, Victoria, Australia

Javaid Iqbal
Department of Mechatronics
College of Electrical and Mechanical Engineering
National University of Sciences and Technology
Islamabad, Pakistan

Muhammad Tahir Khan
Institute of Mechatronics Engineering
University of Engineering and Technology
Peshawar, Pakistan

Farbod Khoshnoud
Department of Mechanical Aerospace and Civil Engineering
College of Engineering, Design and Physical Sciences
Brunel University
Uxbridge, Middlesex, United Kingdom

Haoxiang Lang
Department of Automotive, Mechanical and Manufacturing Engineering
University of Ontario Institute of Technology
Oshawa, Ontario, Canada

Feng Lin
Temasek Laboratories
National University of Singapore
Singapore

Dulini Mendis
Department of Mechanical Engineering
University of Melbourne
Parkville, Victoria, Australia

Diogo Montalvão
School of Engineering and Technology
University of Hertfordshire
Hatfield, Hertfordshire, United Kingdom

William Mortel
Trelleborg Industrial AVS
Leicester, Leicestershire, United Kingdom

Kemao Peng
Temasek Laboratories
National University of Singapore
Singapore

Steven Petrou
Florey Institute for Neuroscience and Mental Health
Centre for Neural Engineering
Department of Electrical and Electronic Engineering
University of Melbourne
Parkville, Victoria, Australia

Leila Pezeshki
Department of Mechanical Engineering
Isfahan University of Technology
Isfahan, Iran

Andi Sudjana Putra
Engineering Design and Innovation Centre
Faculty of Engineering
National University of Singapore
Singapore

Steve Sewell
Trelleborg Industrial AVS
Leicester, Leicestershire, United Kingdom

Kok Kiong Tan
Department of Electrical and Computer Engineering
Faculty of Engineering
National University of Singapore
Singapore

Fei Wang
Temasek Laboratories
National University of Singapore
Singapore

Ying Wang
Department of Mechatronics Engineering
Kennesaw State University
Marietta, Georgia

Bin Xi
Department of Automation
Xiamen University
Xiamen, Fujian, China

Bashan Zuo
Department of Electrical Engineering
Kennesaw State University
Marietta, Georgia

1

Mechatronic Engineering

Clarence W. de Silva

CONTENTS

1.1 Introduction 1
1.2 Modeling and Design 3
1.3 Mechatronic Design Concept 4
    1.3.1 Coupled Design 5
    1.3.2 Mechatronic Design Quotient (MDQ) 7
    1.3.3 Design Evolution 8
1.4 Mechatronic Instrumentation 9
1.5 Evolution of Mechatronics 10
1.6 Application Areas 11
1.7 Conclusion 12
References 12

SUMMARY This chapter introduces the subject of mechatronic engineering. It presents the relevance of modeling, design, and instrumentation in the development of a mechatronic system or product. The evolution of the discipline of mechatronics from its early days leading to the state of the art at present is outlined. Several application areas are given. The concepts of mechatronic design quotient (MDQ) and design evolution through modeling, health monitoring, design expert system, and evolutionary optimization are introduced.

1.1 Introduction

The subject of mechatronics concerns the synergistic application of mechanics, electronics, controls, and computer engineering in the development of electromechanical products and systems through an integrated design approach. Mechatronics is particularly applicable in mixed-domain (or multidomain) systems, which incorporate several physical domains, such as electrical, mechanical, fluid, and thermal, in an integrated manner. For example, the antilock braking system (ABS) of an automobile may involve mechanics, electronics, hydraulics, and heat transfer and may be designed in an "optimal" manner as a mechatronic product. Mechatronic products and systems include modern automobiles and aircraft, smart household appliances, medical robots, space vehicles, and office automation devices.

A typical mechatronic system consists of a mechanical skeleton, actuators, sensors, controllers, signal conditioning/modification devices, computer/digital hardware and software, interface devices, and power sources. Different types of sensing and information acquisition and transfer are involved among all these various types of components. For example, a servomotor, which is a motor with the capability of sensory feedback for accurate generation of complex motions, consists of mechanical, electrical, and electronic components. In a servomotor, the main mechanical components are the rotor, stator, bearings, mechanics of the speed sensor such as an optical encoder, and the motor housing. The electrical components include the circuitry for the field windings and rotor windings (not in the case of permanent-magnet rotors), and circuitry for power transmission and commutation (if needed). Electronic components include those needed for sensing (e.g., an optical encoder for displacement and speed sensing and/or tachometer for speed sensing). As another example, hard disk drives (HDD; see Figure 1.1) of computers use micro-miniature mechanical structures, actuators, and sensors for their operation and control.

The overall design of these devices can be improved by taking a mechatronic approach, with which all components and functions are treated concurrently in an integrated manner in their design. Furthermore, the study of mechatronic engineering should include all stages of modeling, design, development, integration, instrumentation, control, testing, operation, and maintenance of a mechatronic system.

Technology issues of a general mechatronic system are indicated in Figure 1.2. It is seen that they span the traditional fields of mechanical engineering, electrical and electronic engineering, control engineering, and computer engineering. Each aspect or issue within the system may take a multidomain character. For example, an actuator (e.g., DC servomotor) itself may represent a mechatronic device within a larger mechatronic system, such as an automobile or a robot.

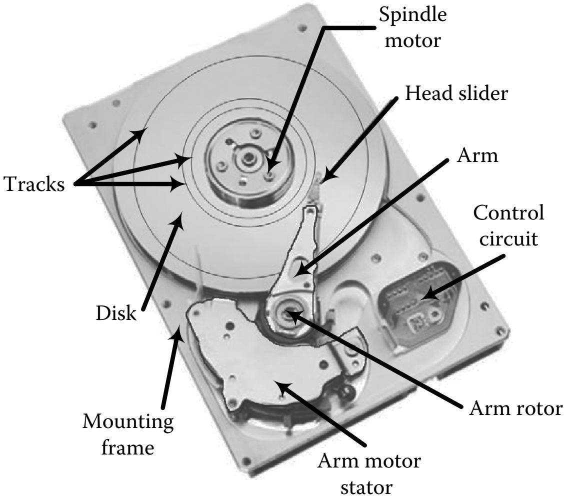


FIGURE 1.1
A HDD unit of a computer.

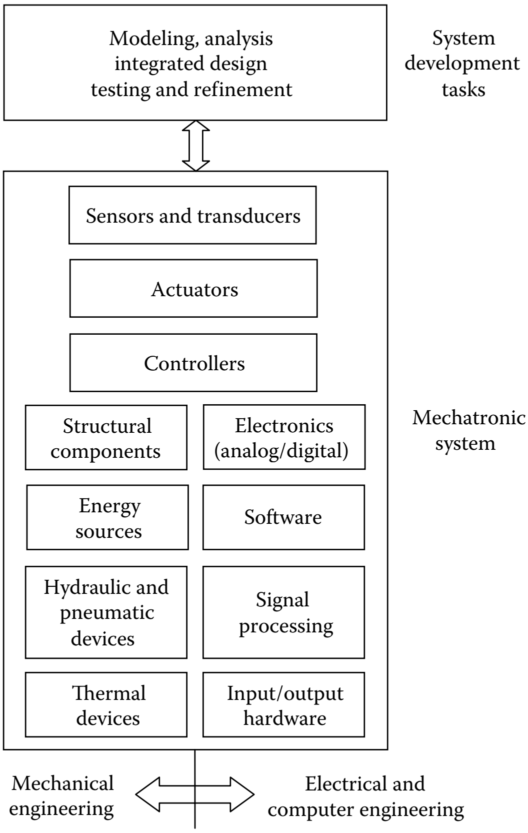

Modeling, analysis
integrated design
testing and refinement
System
development
tasks
Sensors and transducers
Actuators
Controllers
Structural
components
Electronics
(analog/digital)
Energy
sources
Software
Hydraulic and
pneumatic
devices
Signal
processing
Thermal
devices
Input/output
hardware
Mechatronic
system
Mechanical
engineering
Electrical and
computer engineering

FIGURE 1.2
Concepts and technologies of a mechatronic system.

1.2 Modeling and Design

A model is a representation of a real system, and the subject of model development (modeling) is important in mechatronics [1,2]. Modeling and design can go hand in hand in an iterative manner. Of course, in the beginning of the design process, the desired system does not exist. In this context, a model of the anticipated system can be very useful. In view of the complexity of a design process, particularly when striving for an optimal design, it is useful to incorporate system modeling as a tool for design iteration particularly because prototyping can become very costly and time-consuming. Some details are found in Chapters 2, 3, and 9.

In the beginning, by knowing some information about the system (e.g., intended functions, performance specifications, past experience, and knowledge of related systems) and using the design objectives, it is possible to develop a model of sufficient (low to moderate) detail and complexity. By analyzing and carrying out computer simulations of the model, it will be possible to generate useful information that will guide the design process (e.g., generation of a conceptual design or preliminary design). In this manner, design decisions can be made, and the model can be refined using the available (improved) design. This iterative link between modeling and design is schematically shown in Figure 1.3.

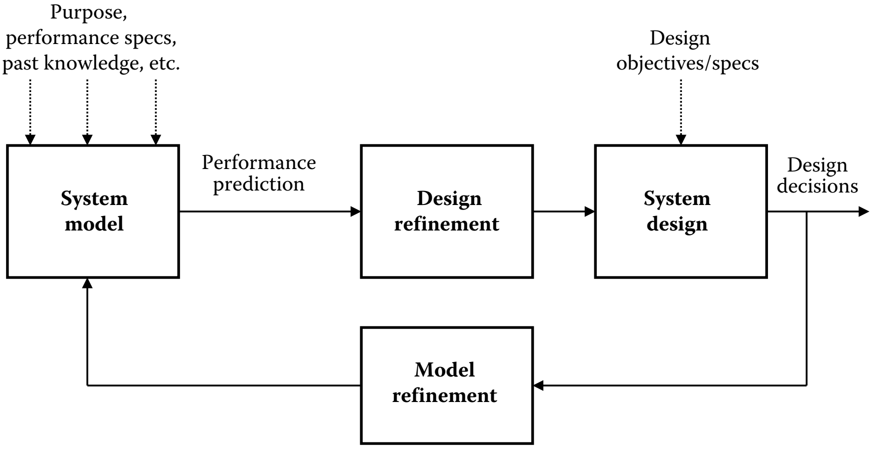


FIGURE 1.3
Link between modeling and design.

It is expected that the mechatronic approach will result in higher quality of products and services, improved performance, and increased reliability while approaching some form of optimality. This will enable the development and production of electromechanical systems efficiently, rapidly, and economically. When performing an integrated design of a mechatronic system, the concepts of energy and power present a unifying thread. The reasons are clear. First, in an electromechanical system, ports of power/energy exist, which link electrical dynamics and mechanical dynamics. Hence, modeling, analysis, and optimization of a mechatronic system can be carried out using a hybrid system (or multidomain system) formulation (a model) that integrates mechanical aspects and electrical aspects of the system. Second, an optimal design will aim for minimal energy dissipation and maximum energy efficiency. There are related implications; for example, greater dissipation of energy will mean reduced overall efficiency and increased thermal problems, noise, vibration, malfunctions, wear and tear, and increased environmental impact. Again, a hybrid model that presents an accurate picture of energy/power flow within the system will present an appropriate framework for the mechatronic design.

A design may use excessive safety factors and worst-case specifications (e.g., for mechanical loads and electrical loads). This will not provide an optimal design or may not lead to the most efficient performance. Design for optimal performance, however, may not necessarily lead to the most economical (least costly) design. When arriving at a truly optimal design, an objective function that takes into account all important factors (performance, quality, cost, speed, ease of operation, safety, environmental impact, etc.) has to be optimized. A complete design process should generate the necessary details for construction or assembly of the system.

1.3 Mechatronic Design Concept

In a true mechatronic sense, the design of a multidomain multicomponent system will require simultaneous consideration and integrated design of all its components as indicated in Figure 1.2. Such an integrated and "concurrent" design will call for a fresh look at the design process itself and also a formal consideration of information and energy transfer between components within the system.

In an electromechanical system, there exists an interaction (or coupling) between electrical dynamics and mechanical dynamics. Specifically, electrical dynamics affect the mechanical dynamics and vice versa. Traditionally, a "sequential" approach has been adopted to the design of multidomain (or mixed) systems, such as electromechanical systems. For example, first, the mechanical and structural components are designed. Next, electrical and electronic components are selected or developed and interconnected. Then, a computer is selected and interfaced with the system; subsequently, a controller is added and so on. The dynamic coupling between various components of a system dictates, however, that an accurate design of the system should consider the entire system as a whole rather than designing the electrical/electronic aspects and the mechanical aspects separately and sequentially. When independently designed components are interconnected, several problems can arise:

1. When two independently designed components are interconnected, the original characteristics and operating conditions of the two will change due to loading or dynamic interactions.
2. Perfect matching of two independently designed and developed components will be practically impossible. As a result, a component can be considerably underutilized or overloaded in the interconnected system, both conditions being inefficient and undesirable.
3. Some of the external variables in the components will become internal and "hidden" due to interconnection, which can result in potential problems that cannot be explicitly monitored through sensing and cannot be directly controlled.

The need for an integrated and concurrent design for electromechanical systems can be identified as a primary motivation for the developments in the field of mechatronics.

1.3.1 Coupled Design

An uncoupled design is one in which each subsystem is designed separately (and sequentially) while keeping the interactions with the other subsystems constant (i.e., ignoring the dynamic interactions). Mechatronic design involves integrated or "coupled" design. The concept of mechatronic design may be illustrated using an example of an electromechanical system, which can be treated as a coupling of an electrical subsystem and a mechanical subsystem. A simplified model for the system is shown in Figure 1.4a. Note that the two subsystems are coupled using a loss-free (pure) energy transformer while the losses (energy dissipation) are integral with the subsystems. In this system, assume that under normal operating conditions the energy flow is from the electrical subsystem to the mechanical subsystem (i.e., the electrical subsystem behaves like a motor rather than a generator). At the electrical port that connects to the energy transformer, there exists a current i (a "through" variable) flowing in and a voltage v (an "across" variable) with the shown polarity [1,2]. The product vi is the electrical power, which is positive out of the electrical subsystem and into the transformer. Similarly, at the mechanical port that comes out of the energy transformer, there exists a torque τ (a through variable) and an angular speed ω (an across variable) with the sign convention indicated in Figure 1.4a. Accordingly,

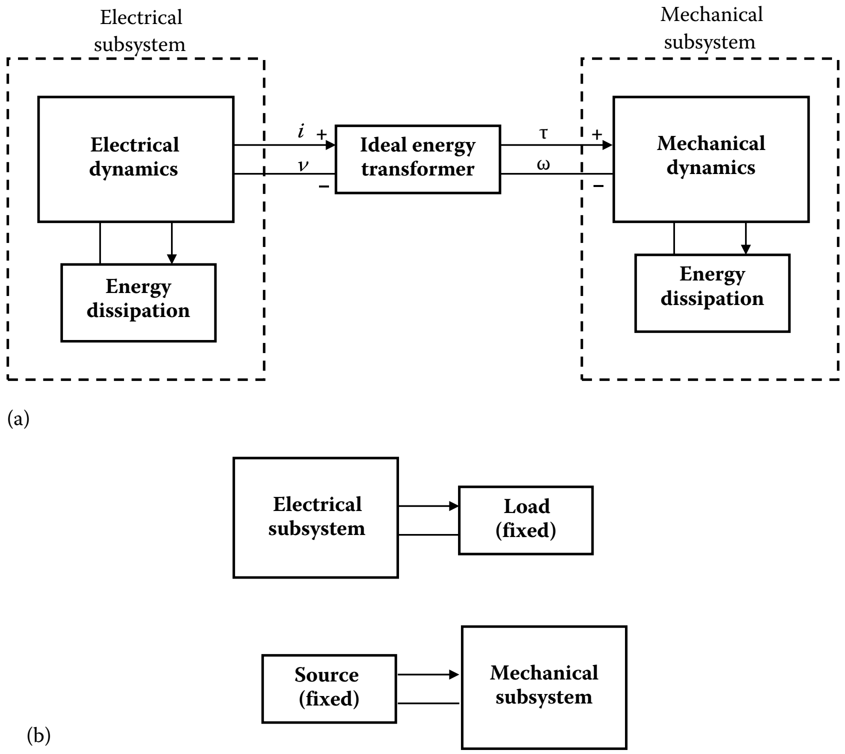

Electrical subsystem
Mechanical subsystem
Electrical dynamics
Energy dissipation
i
+
v
−
Ideal energy transformer
τ
ω
+
−
Mechanical dynamics
Energy dissipation
(a)
Electrical subsystem
Load (fixed)
Source (fixed)
Mechanical subsystem
(b)

FIGURE 1.4
(a) An electromechanical system and (b) conventional design.

a positive mechanical power $\omega\tau$ flows out of the transformer and into the mechanical subsystem. The ideal transformer implies the following:

$$vi = \omega\tau \tag{1.1}$$

In a conventional uncoupled design of the system, the electrical subsystem is designed by treating the effects of the mechanical subsystem as a fixed load, and the mechanical subsystem is designed by treating the electrical subsystem as a fixed energy source as indicated in Figure 1.4b. Suppose that, in this manner, the electrical subsystem achieves an optimal "design index" of I_{ue}, and the mechanical subsystem achieves an optimal design index of I_{um}.

NOTE: The design index is a measure of the degree to which the particular design satisfies the design specifications (design objectives).

When the two uncoupled designs (subsystems) are interconnected, there will be dynamic interactions. As a result, neither the electrical design objectives nor the mechanical design objectives will be satisfied at the levels dictated by I_{ue} and I_{um}, respectively. Instead, they will be satisfied at lower levels as given by the design indices I_e and I_m. A truly mechatronic design will attempt to bring I_e and I_m as close as possible to I_{ue} and I_{um}, respectively. This may be achieved, for example, by minimizing the quadratic cost function:

$$J = \alpha_e(I_{ue} - I_e)^2 + \alpha_m(I_{um} - I_m)^2 \tag{1.2}$$

subject to

$$\begin{bmatrix} I_e \\ I_m \end{bmatrix} = \boldsymbol{D}(\boldsymbol{p}) \tag{1.3}$$

Here, $\boldsymbol{D}$ denotes the transformation that represents the design process, and $\boldsymbol{p}$ denotes information, including system parameters, that is available for the design.

Even though this formulation of the mechatronic design problem appears rather simple and straightforward, the reality is otherwise. In particular, the design process, as denoted by the transformation $\boldsymbol{D}$, can be quite complex and typically nonanalytic. Furthermore, minimization of the cost function J is by and large an iterative practical scheme and undoubtedly a knowledge-based and nonanalytic procedure. This complicates the process of mechatronic design. In any event, the design process will need the information represented by $\boldsymbol{p}$.

1.3.2 Mechatronic Design Quotient (MDQ)

Mechatronic systems are complex and require multiple technologies in multiple domains. Their optimal design may call for multiple performance indices. The problem of mechatronic design may be treated as a maximization of a "mechatronic design quotient" or MDQ, the concept of which was introduced by de Silva in 2003 [3]. In particular, an alternative formulation of the optimization problem given by Equations 1.2 and 1.3 would be the maximization of the mechatronic design quotient:

$$MDQ = \frac{\alpha_e I_e^2 + \alpha_m I_m^2}{\alpha_e I_{ue}^2 + \alpha_m I_{um}^2} \tag{1.4}$$

subject to Equation 1.3.

Even though Equation 1.4 is formulated for two categories of technologies or devices m and e (and the corresponding indices I_e and I_m), the MDQ may be generalized for three or more categories, such as reliability, maintainability, efficiency, cost-effectiveness, power and efficiency, size and geometry, control friendliness, and level of intelligence. The corresponding indices may be qualitative or nonanalytic and may have correlations or interactions. Then, more sophisticated representations (e.g., the use of fuzzy measures) and optimization techniques (e.g., evolutionary computing or genetic programming or GP) may be employed in the design process.

For example, in the use of genetic algorithms (GA) for mechatronic design, we start with a group (population) of initial chromosomes (embryos) in which an individual chromosome is one possible design. An individual gene in a chromosome corresponds to an element of information in a design (e.g., system component, connection structure, set of parameters, design attribute). Alleles are possible values of a gene (e.g., available choices for a particular component). The "fitness function" of the GA represents the "value" or "goodness" or "fitness" of a design. In the present context, the fitness function is the MDQ, which is computable for a given design once the element information of the design is known). Then the problem of design optimization becomes

$$\text{Maximize } MDQ(p_1, p_2, \ldots p_m) \tag{1.5}$$

where p_i = ith design aspect.

The strength and applicability of the MDQ approach stem from the possibility that the design process may be hierarchically separated. Then, an MDQ may be optimized for one design layer involving two more technology groups in that layer before proceeding to the next lower design layer on which each technology group is separately optimized by considering several technology/component groups within that group together with an appropriate MDQ for that lower-level design problem. For example, an upper layer may optimize the actuator type for the particular application (e.g., hydraulic, DC, induction, stepper) with an appropriate MDQ. The next lower level may optimize the motor selection (e.g., select a motor from an available set of DC motors) with another MDQ. In this manner, a complex design optimization may be achieved through several design optimizations at different design levels. The final design may not be precisely optimal yet intuitively adequate for practical purposes, say in a conceptual design. The concept of MDQ has been enhanced and applied in subsequent work [4–9]. Further details are found in Chapter 9.

1.3.3 Design Evolution

Traditionally, online monitoring of responses or outputs of a system may be used to detect and diagnose the faults and malfunctions (existing or impending) of a system. We believe such monitoring may also be used to improve the design of an existing mechatronic system. In particular, just like how a health monitoring system can pinpoint a defective component in a system, it should be possible for the same system to at least identify the possible regions (sites) of design weakness in the system. This is the premise of the approach for “design evolution” as outlined below.

A model of the existing system (whose design needs to be improved) and evolutionary computing (genetic programming) may facilitate the approach of “evolutionary” design improvement through online monitoring. A possible framework for implementing this approach is indicated in Figure 1.5.

The relevant steps are as follows:

1. Develop a model of the existing system.
2. Establish (using a machine health monitoring system and an expert system) which aspects or segments of the original system (and its model) may be modified or improved using information monitored from the system. These will provide “modifiable sites” for the existing system or model.
3. Formulate a performance function to represent the “goodness” of the design. This is the MDQ.
4. Use an optimization method (e.g., GP) to evolve the model so as to maximize the performance function.
5. Implement in the existing system the design changes represented by the evolved model.

The optimization scheme will gradually improve the original model of the system so as to produce better performance (as judged by the MDQ). This will require comparison of the monitored response of the original system and the simulated response of the model as it evolves (improves) with respect to the MDQ. In Figure 1.5, in addition to the initial model of the system, evolutionary computing approach, and online monitoring, we have shown an expert system as well for “intelligent” decision making associated with design or model

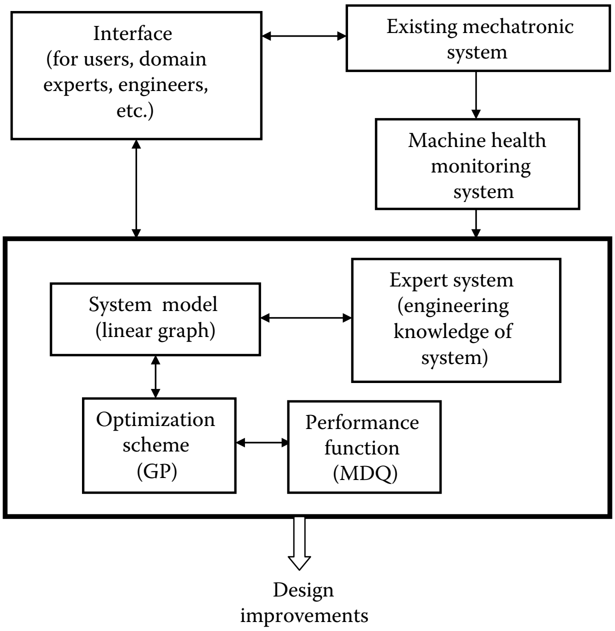

Interface
(for users, domain experts, engineers, etc.)
Existing mechatronic system
Machine health monitoring system
System model
(linear graph)
Expert system
(engineering knowledge of system)
Optimization scheme
(GP)
Performance function
(MDQ)
Design improvements

FIGURE 1.5
Structure of a system for evolutionary design.

improvement. This expert system may be generated from the knowledge or expertise of the existing system, its design, and engineering know-how. The concept of evolutionary design as represented in Figure 1.5 was first presented by de Silva in a 2007 proposal [10]. It has been further investigated and enhanced in subsequent work [11,12].

1.4 Mechatronic Instrumentation

From the viewpoint of mechatronics, a somewhat "optimal" and unified approach, not a sequential approach, has to be taken in the instrumentation process with regard to sensors, actuators, interface hardware, and controllers. Specifically, "instrumentation" has to be treated as an integral aspect of "design." This is justified because through design we develop systems that can carry out required functions while meeting certain performance specifications [13]. Sensors, actuators, control, and instrumentation play a direct role in achieving the design objectives. See Chapters 4 through 6 for further details.

Even though, in theory, the mechatronic approach is the "best" ("optimal"), particularly with regard to instrumentation, it may not be practical to realize the optimal results of instrumentation as dictated by the approach. The mechatronic approach requires the entire system, including the process and instrumentation, to be designed concurrently. This assumes that all aspects and components of the entire system can be modified according to the mechatronic result. However, unless the entire system (including the process) is a new design, such flexibility is often not realistic. For example, typically, the process is already available, and it is not practical, convenient, or cost-effective to modify some or all of its components. Then, even if the instrumentation is chosen according to a mechatronic procedure, the overall system will not function as optimally as if we had the freedom to modify the entire plant as well.

As an example, consider an automated vehicle guideway system of public transit. Suppose that the system already exists and that it is required to replace some of its cars. Then, it is not practical to significantly modify the guideway to accommodate a new design of cars. In fact, the design freedom with regard to the cars will be limited even if it is not constrained by a specific guideway design. If a car is designed and instrumented optimally, according to mechatronics, the operation of the overall vehicle guideway system will not be optimal.

It is clear that true mechatronic instrumentation may not be possible for existing processes. Furthermore, because the components for instrumentation (sensors, actuators, controllers, accessory hardware) may come from different manufacturers, and their availability would be limited, it is not practical to realize a true "mechatronic" product (because the available set of components is limited and may not also be truly compatible).

1.5 Evolution of Mechatronics

Mechanical engineering products and systems that employ some form of electrical engineering principles and devices have been developed and used since the early part of the 20th century. These systems included the automobile, electric typewriter, aircraft, and the elevator. Some of the power sources used in these systems were not necessarily electrical, but there were batteries and/or conversion of thermal power into electricity through generators. These "electromechanical" systems were not "mechatronic" systems because they did not use an integrated approach characterizing mechatronics for their analysis, design, development, and implementation.

Rapid advances in electromechanical devices and systems were possible particularly due to developments in control engineering, which began for the most part in the early 1950s, and still more rapid advances in digital computer and communication as a result of integrated circuit (IC) and microprocessor technologies, starting from the late 1960s. With these advances, engineers and scientists felt the need for a multidisciplinary approach to design and hence a "mechatronic" approach. In 1969, Yasakawa Electric in Japan was the first to coin the term "mechatronics," for which the company obtained a trademark in 1972. Subsequently, in 1982, the company has released the trademark rights. Even though a need for mechatronics was felt even in those early times, no formal discipline and educational programs existed for engineers to be educated and trained in this area. The research and development activities mainly in automated transit systems and robotics in the 1970s and 1980s undoubtedly paved the way for the evolution of the field of mechatronics. With today's sophisticated technologies of mechanics and materials, analog and digital electronics, sensors, actuators, controllers, electromechanical design, and microelectromechanical systems (MEMS) with embedded sensors, actuators, and microcontrollers, the field of mechatronics has attained a high degree of maturity. Now many universities around the world offer undergraduate and graduate programs in mechatronic engineering, which have become highly effective and popular among students, instructors, employees, and employers alike.

1.6 Application Areas

Application areas of mechatronics are numerous and involve those that concern multidomain (mixed) systems and particularly electromechanical systems. These applications may involve the following:

1. Modifications and improvements to conventional designs by using a mechatronic approach
2. Development and implementation of original and innovative mechatronic systems

In either category, the applications may employ sensing, actuation, control, signal conditioning, component interconnection and interfacing, and communication, generally using tools of mechanical, electrical and electronic, computer, and control engineering. Some important areas of application are indicated here.

Transportation is a broad area in which mechatronic engineering has numerous applications. In ground transportation in particular, automobiles, trains, and automated transit systems use mechatronic devices. They include airbag deployment systems, ABS, cruise control systems, active suspension systems, and various devices for monitoring, toll collection, navigation, warning, and control in intelligent vehicular highway systems (IVHS). In air transportation, modern aircraft designs with advanced materials, structures, electronics, and control benefit from the concurrent and integrated approach of mechatronics to develop improved designs of flight simulators, flight control systems, navigation systems, landing gear mechanisms, traveler comfort aids, and the like.

Manufacturing and production engineering is another broad field that uses mechatronic technologies and systems. Factory robots (for welding, spray-painting, assembly, inspection, and so on), automated guided vehicles (AGVs), modern computer numerical control (CNC) machine tools, machining centers, rapid (and virtual) prototyping systems, and micromachining systems are examples of mechatronic applications. High-precision motion control is particularly important in these applications.

In medical and healthcare applications, robotic technologies for patient examination, surgery, rehabilitation, drug dispensing, telemedicine, and general patient care are being developed and used. Mechatronic technologies are being applied for patient transit devices, various diagnostic probes and scanners, beds, and exercise machines.

In a modern office environment, automated filing systems, multifunctional copying machines (copying, scanning, printing, fax, and so on), food dispensers, multimedia presentation and meeting rooms, and climate-control systems incorporate mechatronic technologies.

In household applications, home security systems, robotic caregivers and helpers, robotic vacuum cleaners, washers, dryers, dishwashers, garage door openers, and entertainment centers use mechatronic devices and technologies.

The computer industry can considerably benefit from mechatronics. The impact goes further because digital computers are integrated into a vast variety of other devices and applications.

In civil engineering applications, cranes, excavators, and other machinery for building, earth removal, mixing, and so on, will improve their performance by adopting a mechatronic design approach.

In space applications, mobile robots such as NASA's Mars exploration Rover, space-station robots, and space vehicles are fundamentally mechatronic systems.

It is noted that there is no end to the type of devices and applications that can incorporate mechatronics. In view of this, the traditional boundaries between engineering disciplines will become increasingly fuzzy, and the field of mechatronics will grow and evolve further through such merging of disciplines. Also, see Chapters 8, 10–13, and 15.

1.7 Conclusion

This chapter introduced the subject of mechatronic engineering. It presented the relevance of modeling, design, and instrumentation in the development of a mechatronic system or a product. The evolution of the discipline of mechatronics from its early days leading to the state of the art at present was outlined. Several application areas were given. The concepts of MDQ and design evolution through modeling, health monitoring, design expert system, and evolutionary optimization were introduced.

References

1. De Silva, C. W., *Modeling and Control of Engineering Systems*, Taylor & Francis, CRC Press, Boca Raton, FL, 2009.
2. De Silva, C. W., *Mechatronics—A Foundation Course*, Taylor & Francis, CRC Press, Boca Raton, FL, 2010.
3. De Silva, C. W., "Sensing and Information Acquisition for Intelligent Mechatronic Systems," *Science and Technology of Information Acquisition and Their Applications, Proceedings of the Symposium on Information Acquisition*, Chinese Academy of Sciences, Hefei, China, pp. 9–18, November 2003 (Invited, Keynote Address).
4. Behbahani, S. and de Silva, C. W., "Mechatronic Design Quotient (MDQ) as the Basis of a New Multi-Criteria Mechatronic Design Methodology," *IEEE/ASME Transactions on Mechatronics*, Vol. 12, No. 2, pp. 227–232, 2007.
5. Sun, J., Poo, A. N., Ang, M. H., Hong, G. S., de Silva, C. W., and Tan, K. K., "Design of Vibration Controllers for Flexible Beams Using the Mechatronic Design Quotient (MDQ) Approach," *Journal of Vibration and Control*, Vol. 13, No. 1, pp. 65–94, 2007.
6. Behbahani, S. and de Silva, C. W., "System-based and Concurrent Design of a Smart Mechatronic System Using the Concept of Mechatronic Design Quotient (MDQ)," *IEEE/ASME Transactions on Mechatronics*, Vol. 13, No. 1, pp. 14–21, 2008.
7. Sun, J., Poo, A. N., de Silva, C. W., Chew, C. M., Ang, M. H., Tan, K. K., and Hong, G. S., "Optimization of Mechatronic Design Quotient (MDQ) Using Genetic Algorithm in Vibration Controllers for Flexible Beams," *Journal of Vibration and Control*, Vol. 15, No. 11, pp. 1603–1626, 2009.
8. De Silva, C. W. and Behbahani, S., "A Design Paradigm for Mechatronic Systems," *Mechatronics*, Vol. 23, No. 8, pp. 960–966, 2013.
9. Behbahani, S. and de Silva, C. W., "Mechatronic Design Evolution Using Bond Graphs and Hybrid Genetic Algorithm with Genetic Programming," *IEEE/ASME Transactions on Mechatronics*, Vol. 18, No. 1, pp. 190–199, 2013.

10. De Silva, C. W., "A Unified System Framework Including Intelligent Supervisory Control and On-Line Design Evolution with Networked Remote Operation," *A Proposal for a Tier 1 Canada Research Chair*, Natural Sciences and Engineering Research Council, Ottawa, Canada, 2008.
11. Gamage, L. B. and de Silva, C. W., "A System Framework with On-line Monitoring and Evaluation for Design Evolution of Engineering System," *ASME Journal of Computing and Information Science in Engineering*, Vol. 10, pp. 1–6, 2010.
12. Gamage, L. B., de Silva, C. W., and Campos, R. "Design Evolution of Mechatronic Systems through Modeling, On-line Monitoring, and Evolutionary Optimization," *Mechatronics*, Vol. 22, No. 1, pp. 83–94, 2012.
13. De Silva, C. W., *Sensors and Actuators—Engineering System Instrumentation*, 2nd Edition, Taylor & Francis, CRC Press, Boca Raton, FL, 2015.

Section I

Fundamentals

2

Modeling for Control of Rigid Bodies in 3-D Space

Ibrahim Esat, Minoo Dabestani, William Mortel, and Steve Sewell

CONTENTS

2.1 Introduction 18
2.2 Theory 19
  2.2.1 Definitions and Assumptions 19
  2.2.2 Equations of Motion for the Linear Model 19
  2.2.3 Linear Momentum Force Systems 19
    2.2.3.1 Stiffness and Damping Systems 19
    2.2.3.2 Generalization of the Equation of Linear Momentum 20
  2.2.4 Generalization of the Equations of Moment of Momentum 21
  2.2.5 Assembly of Equations 21
2.3 Modeling Sensors and Actuators into the Model 26
  2.3.1 Modeling Actuators 27
  2.3.2 Modeling Sensors and Feedback 30
2.4 Introduction to Software MBDS 32
  2.4.1 A Simple Two-Mass Spring System with an Actuator and a Relative Velocity Sensor 33
  2.4.2 Response of the System to a Simple Step Function 33
2.5 Conclusions 34
References 35

SUMMARY This chapter gives a full derivation of the dynamics and control of multiple rigid bodies in a 3-D space. Almost all existing control literature concentrates on point mass systems. Although there is wide coverage of multiple degrees of freedom in the literature, they are not representative of realistic situations. Modeling of the dynamics of rigid bodies presents difficulties as far as the modeling of spring and damper elements are concerned. The same difficulties are also related to the actuator elements. A further complication is due to the fact that actuator dynamics are influenced by the kinematics of their assembly points as well as feedback from sensors. What a sensor measures in a rigid body environment is the point motion on a rigid body, which is made of up to six degrees of motion relating to the rigid body, or if it is measuring motion between two moving bodies, then 12 degrees are involved. What is measured sometimes is in one direction (a typical accelerometer); therefore, there is no possibility of knowing state variables that generate such motion uniquely. Not having a unique relationship to state variables complicates the feedback control strategy. All these considerations make rigid body control a challenging task. The present chapter deals with these issues and offers a complete mathematical treatment to the problem.

2.1 Introduction

The modeling presented here is based on the Euler Newton formulation. Multiple bodies are interconnected by discrete springs and/or dashpots. In formulation, the connections are described in terms of end coordinates of springs relative to the axes system of the body to which it is attached. However, in software implementation, all positions are relative to the global axes. Stiffness characteristics are described along the three principal axes of springs. Orientation of springs and masses is described by using the Euler angles. The model developed is geometrically linear, and the vibration amplitudes are small; it means that the orientation of bodies may be assumed to be approximately constant. It is also assumed that the oscillation does not result in accelerations due to second-order velocities. Also, for what is presented here, it is also assumed that the springs are linear although MBDS can perform nonlinear analysis for both time domain simulation and time domain control. The model includes actuators for control purposes, and although the actuators are assumed to be of the electromagnetic type, modeling given can easily be modified for any other type of actuators. Actuators are similar to the springs, and dashpots are discrete with small actuations in order not to violate the geometrical linearity of the system. Sensors are located at arbitrary locations (locations that may be different than the locations of the actuators). Sensors are used to provide feedback, which is used to control the input voltage to the actuators. Therefore, the actuator force is a function of the kinematics of its connection points and the feedback from the sensors.

This chapter gives a detailed treatment of the formulation of equations of motion of rigid bodies in 3-D space; such a system is shown in Figure 2.1. This is accomplished by starting with the assumptions and progressing step by step, first treating the stiffness element and its matrix representation when it is attached at an arbitrary point on the body. Mass elements are also taken through similar and clear step-by-step formulations, giving the final matrix representation.

As well as presenting full formulation and assembly, issues relating to the programming aspect of the work are also presented. The final section presents MBDS, a software suite, developed based on the formulation presented in this chapter.

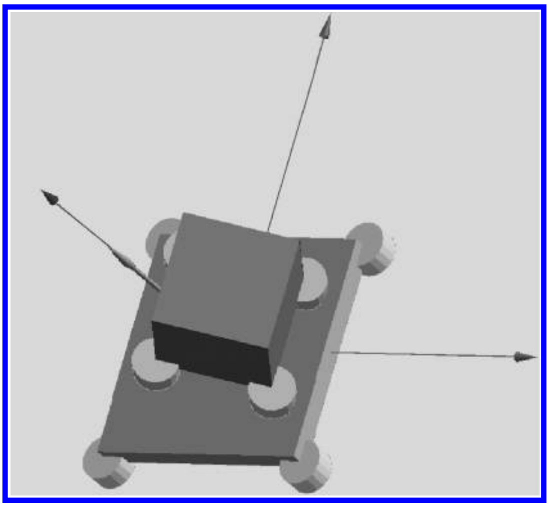


FIGURE 2.1
A typical multibody system copied from a MBDS screen.

2.2 Theory

2.2.1 Definitions and Assumptions

i. It is assumed that the springs have zero length.
ii. It is assumed that the stiffness of the springs in their principal axes of deflection remains uncoupled.
iii. Amplitude of oscillation is small. No geometrical nonlinearity is involved. In other words, the orientation of both mountings and bodies remain unaffected by oscillations.
iv. The time-dependent effects of polymeric material are excluded.
v. Gyroscopic effects are assumed to be small and negligible.

These assumptions are acceptable for most of the engineering vibration problems with small amplitude vibration.

2.2.2 Equations of Motion for the Linear Model

To set up equations of motion for a dynamic system, the following are required:

i. Generalization of the equations of internal reactions and external forces. The internal reactions due to damping and stiffness elements have to be expressed in a unified and structured fashion for formulation of the stiffness matrix. (Damping matrix structure is identical to stiffness matrix structure except that stiffness coefficients need to be replaced by damping coefficients.)
ii. Generalization of the equations of linear momentum (force–acceleration equations).
iii. Generalization of the equations of angular momentum (turning moment equations).

2.2.3 Linear Momentum Force Systems

2.2.3.1 Stiffness and Damping Systems

The formulations carried out in this chapter for obtaining the stiffness matrix equally apply to the damping matrix by replacing stiffness parameters with their corresponding damping parameters. Let us assume that spring stiffnesses are described in a local 3-D Cartesian frame system, the axes of which coincide with the principal axes of the spring. The force vector **f** acting on the spring may be expressed as

$$\mathbf{f} = \mathbf{k}\mathbf{x} \tag{2.1}$$

Here **k** is the stiffness matrix (diagonal containing principal stiffness values), and **x** is the displacement vector (expressing extension in the spring).

In general, it is convenient to describe the behavior of a system in the global axes frame, OXYZ. This is not a prerequisite for the formulation; it is equally possible to obtain equations of motion of each body in its own frame. In this chapter, all spring stiffness matrices will be expressed in a common global axes frame. Therefore, individual spring matrices will be transformed accordingly. Because the principal axes of spring and the global axes are all orthogonal, an orthogonal transformation exists between the two frames. A vector

x in the local axes coordinates could be expressed as a vector **X** in the global axes system. Using **T**, a transformation matrix,

$$\mathbf{X} = \mathbf{Tx} \tag{2.2}$$

Premultiply Equation 2.1 by **T**,

$$\mathbf{Tf} = \mathbf{Tkx}, \tag{2.3}$$

But

$$\mathbf{Tf} = \mathbf{F} \tag{2.4}$$

Therefore, force vector **F** in the global axis frame, may be written as

$$\mathbf{F} = \mathbf{Tkx} \tag{2.5}$$

For consistency, **x** needs to be replaced by **X**. To replace **x** by **X**, Equation 2.2 may be used:

$$\mathbf{x} = \mathbf{T}^T\mathbf{X} \tag{2.6}$$

This is true because $\mathbf{T}^{-1} = \mathbf{T}^T$ for orthogonal transformation matrices; therefore,

$$\mathbf{F} = \mathbf{TkT}^T\mathbf{X} \tag{2.7}$$

Here,

$$\mathbf{K} = \mathbf{TkT}^T \tag{2.8}$$

where $\mathbf{TkT}^T$ is the stiffness matrix of the spring in the global axes system. The transformation matrix **T** may be described in three Euler angle rotations.

2.2.3.2 Generalization of the Equation of Linear Momentum

If the mass/inertia matrix in Euler-Newton formulation is obtained relative to the axes passing through the center of mass, then the subsection of the mass matrix corresponding to linear momentum is a diagonal matrix containing the mass elements. This is

$$\mathbf{h}_l = \mathbf{m}_i\mathbf{v} \tag{2.9}$$

where $\mathbf{h}_l$ is linear momentum, $\mathbf{m}_i$ is a diagonal matrix, and **v** is the velocity vector of the center of mass (casually known as the center of gravity, COG) of the body.

The usual transformation to the global axis frame, $\mathbf{H}_l = \mathbf{TmT}^T\mathbf{v}$, leaves the mass matrix **m** unchanged. Therefore, the force acting on a body *i* (i.e., the rate of change of linear momentum) may be expressed simply as

$$Force = \dot{\mathbf{H}}_l = \frac{\partial \mathbf{H}_l}{\partial t} = \mathbf{ma} \tag{2.10}$$

Here, **a** is the acceleration vector of COG.

2.2.4 Generalization of the Equations of Moment of Momentum

Similarly, the equations of moment of momentum may be expressed as

$$\mathbf{h}_{\mathrm{a}} = \mathbf{j}\,\boldsymbol{\omega} \tag{2.11}$$

where $\mathbf{h}_{\mathrm{a}}$ is angular momentum vector, $\mathbf{j}$ is the moments of inertia matrix, and $\boldsymbol{\omega}$ is the axes rotation (in this case, axes are attached to the body).

Here, $\mathbf{j}$ may or may not be a diagonal matrix; however, it is always symmetric. Equation 2.7 is described in the local axes system of the rigid body, and it has to be expressed in the global axis system for the final matrix assembly. As presented for the stiffness elements, transformation follows exactly the same steps; obviously, in this case $\mathbf{T}$ refers to the transformation matrix of mass relative to the global axis system. Transforming Equation 2.7 to the global axes,

$$\mathbf{H}_{\mathrm{a}} = \mathbf{T}\,\mathbf{j}\,\mathbf{T}^{T}\,\Omega \tag{2.12}$$

Introducing a new matrix notation,

$$\mathbf{J} = \mathbf{T}\,\mathbf{j}\,\mathbf{T}^{T} \tag{2.13}$$

The vector differentiation of $\mathbf{H}_a$ gives the moment vector in the global axes,

$$M = \dot{\mathbf{H}}_a = \frac{\partial \mathbf{H}_a}{\partial t} + \omega \times \mathbf{H}_a \tag{2.14}$$

Here, $\boldsymbol{\omega}$ is the rotation of the body (or the axes frame as the body is fixed to the axes).

$\boldsymbol{\omega} \times \mathbf{H}_a$ contains the product of angular velocity terms, and this, for small and geometrically linear vibration problems, is small and may be ignored.

2.2.5 Assembly of Equations

To assemble the equations of motion, the internal forces acting on individual bodies due to their motion relative to each other are required. In Figure 2.2, two bodies (**i** and **j**) in motion are shown.

Motion of the origin of axes of system **i** (which coincide with the COG of body **i**) is given by $\mathbf{x}_{\mathrm{i}} = (x_{\mathrm{i}}, y_{\mathrm{i}}, z_{\mathrm{i}})$, and the angular rotation of axes is given by $\boldsymbol{\alpha}_{\mathrm{i}} = (\alpha_{\mathrm{i}}, \beta_{\mathrm{i}}, \gamma_{\mathrm{i}})$. Similarly, the motion of body **j** is described by $\mathbf{x}_{\mathrm{j}} = (x_{\mathrm{j}}, y_{\mathrm{j}}, z_{\mathrm{j}})$ and $\boldsymbol{\alpha}_{\mathrm{j}} = (\alpha_{\mathrm{j}}, \beta_{\mathrm{j}}, \gamma_{\mathrm{j}})$.

For small motion, displacements of end points of springs on each body, described in **the axes frame** of each body, are given by

$$\mathbf{d}_{\mathrm{pi}} = \mathbf{x}_{\mathrm{i}} + \theta_{\mathrm{i}} \times \mathbf{r}_{\mathrm{pi}} \tag{2.15}$$

$$\mathbf{d}_{\mathrm{pi}} = \mathbf{x}_{\mathrm{i}} + \theta_{\mathrm{i}} \times \mathbf{r}_{\mathrm{pi}} \tag{2.16}$$

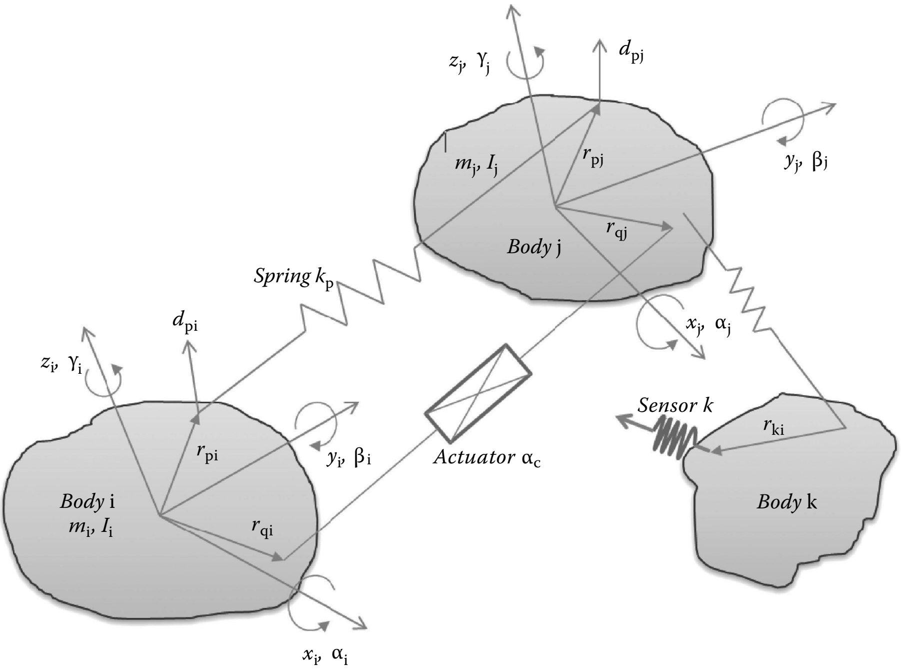

z_j, γ_j
d_{pj}
m_j, I_j
r_{pj}
y_j, β_j
r_{qj}
Body j
Spring k_p
x_j, α_j
d_{pi}
z_i, γ_i
Sensor k
r_{ki}
r_{pi}
y_i, β_i
Actuator α_c
Body i
m_i, I_i
r_{qi}
Body k
x_i, α_i

FIGURE 2.2
Body **i** and body **j** are connected by spring k_p and actuator p_p and sensor **k** placed on body **k**.

Here, r_{pi} and r_{pj} are coordinates of spring attachment relative to bodies i and j in their respective axes frames, given as, $r_{pi} = (x_{pi}, y_{pi}, z_{pi})$ and $r_{pj} = (x_{pj}, y_{pj}, z_{pj})$.

Cross product terms in Equations 2.15 and 2.16 can be converted into matrix form as

$$\mathbf{d}_{pi} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{pi} & -y_{pi} \\ 0 & 1 & 0 & -z_{pi} & 0 & x_{pi} \\ 0 & 0 & 1 & y_{pi} & -x_{pi} & 0 \end{bmatrix} \begin{Bmatrix} x_{oi} \\ y_{oi} \\ z_{oi} \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \end{Bmatrix}$$

$$\mathbf{d}_{pj} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{pj} & -y_{pj} \\ 0 & 1 & 0 & -z_{pj} & 0 & x_{pj} \\ 0 & 0 & 1 & y_{pj} & -x_{pj} & 0 \end{bmatrix} \begin{Bmatrix} x_{oj} \\ y_{oj} \\ z_{oj} \\ \theta_{xj} \\ \theta_{yj} \\ \theta_{zj} \end{Bmatrix} \tag{2.17}$$

Alternatively, notation $\mathbf{d}_\mathrm{i}$ and $\mathbf{d}_\mathrm{j}$ can also be described as

$$\mathbf{d}_\mathrm{pi} = \mathbf{I}\mathbf{x}_\mathrm{i} + \mathbf{R}_\mathrm{pi}\,\theta_\mathrm{i} \tag{2.18}$$

$$\mathbf{d}_\mathrm{pj} = \mathbf{I}\mathbf{x}_\mathrm{j} + \mathbf{R}_\mathrm{pj}\,\theta_\mathrm{j} \tag{2.19}$$

where $\mathbf{x}_\mathrm{i}^T = \{x_\mathrm{oi}\ y_\mathrm{oi}\ z_\mathrm{oi}\}$ and $\boldsymbol{\theta}_\mathrm{i}^T = \{\theta_\mathrm{xi}\ \theta_\mathrm{yi}\ \theta_\mathrm{zi}\}$, and $\mathbf{d}_\mathrm{i}$ and $\mathbf{I}$ in the matrix form are given by

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_\mathrm{pi} = \begin{bmatrix} 0 & z_\mathrm{pi} & -y_\mathrm{pi} \\ -z_\mathrm{pi} & 0 & x_\mathrm{pi} \\ y_\mathrm{pi} & -x_\mathrm{pi} & 0 \end{bmatrix} \tag{2.20}$$

The matrix notations of the displacement $\mathbf{d}_\mathrm{j}$ can be obtained by replacing the suffix i with j. These matrix and vector notations can be combined as $\mathbf{A}_\mathrm{pi}$ and $\mathbf{u}_\mathrm{i}$ for displacement $\mathbf{d}_\mathrm{i}$ and $\mathbf{A}_\mathrm{pj}$ and $\mathbf{u}_\mathrm{j}$ for displacement $\mathbf{d}_\mathrm{j}$ as

$$\mathbf{A}_\mathrm{pi} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_\mathrm{pi} & -y_\mathrm{pi} \\ 0 & 1 & 0 & -z_\mathrm{pi} & 0 & x_\mathrm{pi} \\ 0 & 0 & 1 & y_\mathrm{pi} & -x_\mathrm{pi} & 0 \end{bmatrix} \quad \mathbf{A}_\mathrm{pj} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_\mathrm{pj} & -y_\mathrm{pj} \\ 0 & 1 & 0 & -z_\mathrm{pj} & 0 & x_\mathrm{pj} \\ 0 & 0 & 1 & y_\mathrm{pj} & -x_\mathrm{pj} & 0 \end{bmatrix} \tag{2.21}$$

$$\mathbf{u}_\mathrm{i} = \begin{Bmatrix} x_\mathrm{oi} \\ y_\mathrm{oi} \\ z_\mathrm{oi} \\ \theta_\mathrm{xi} \\ \theta_\mathrm{yi} \\ \theta_\mathrm{zi} \end{Bmatrix} \quad \mathbf{u}_\mathrm{j} = \begin{Bmatrix} x_\mathrm{oj} \\ y_\mathrm{oj} \\ z_\mathrm{oj} \\ \theta_\mathrm{xj} \\ \theta_\mathrm{yj} \\ \theta_\mathrm{zj} \end{Bmatrix} \tag{2.22}$$

Thus, $\mathbf{d}_\mathrm{pi}$ and $\mathbf{d}_\mathrm{pj}$ can be expressed in their new forms as

$$\mathbf{d}_\mathrm{pi} = \mathbf{A}_\mathrm{pi}\mathbf{u}_\mathrm{i} \tag{2.23}$$

$$\mathbf{d}_\mathrm{pj} = \mathbf{A}_\mathrm{pj}\mathbf{u}_\mathrm{j} \tag{2.24}$$

Forces and moments acting on body **i** at position **p** may be expressed as

$$\begin{Bmatrix} F_{\text{pix}} \\ F_{\text{piy}} \\ F_{\text{piz}} \\ T_{\text{pix}} \\ T_{\text{piy}} \\ T_{\text{piz}} \end{Bmatrix} \text{ where } \begin{Bmatrix} T_{\text{pix}} \\ T_{\text{piy}} \\ T_{\text{piz}} \end{Bmatrix} = \mathbf{r}_{\text{pi}} \times \mathbf{F}_{\text{pi}} \text{ and } \mathbf{F}_{\text{pi}} = \begin{Bmatrix} F_{\text{pix}} \\ F_{\text{piy}} \\ F_{\text{piz}} \end{Bmatrix} \tag{2.25}$$

Then, Equation 2.25 may be written in matrix form as

$$\begin{Bmatrix} T_{\text{pix}} \\ T_{\text{piy}} \\ T_{\text{piz}} \end{Bmatrix} = \mathbf{r}_{\text{pi}} \times \mathbf{F}_{\text{pi}} = \begin{bmatrix} 0 & -z_{\text{pi}} & y_{\text{pi}} \\ z_{\text{pi}} & 0 & -x_{\text{pi}} \\ -y_{\text{pi}} & x_{\text{pi}} & 0 \end{bmatrix} \begin{Bmatrix} F_{\text{pix}} \\ F_{\text{piy}} \\ F_{\text{piz}} \end{Bmatrix} \tag{2.26}$$

Therefore,

$$\begin{Bmatrix} F_{\text{pix}} \\ F_{\text{piy}} \\ F_{\text{piz}} \\ T_{\text{pix}} \\ T_{\text{piy}} \\ T_{\text{piz}} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -z_{\text{pi}} & y_{\text{pi}} \\ z_{\text{pi}} & 0 & -x_{\text{pi}} \\ -y_{\text{pi}} & x_{\text{pi}} & 0 \end{bmatrix} \begin{Bmatrix} F_{\text{pix}} \\ F_{\text{piy}} \\ F_{\text{piz}} \end{Bmatrix} \tag{2.27}$$

Note that the matrix in Equation 2.27 is the transposition of the matrix introduced in Equation 2.21. The force equation is written as

$$\mathbf{f}_{\text{pi}} = \mathbf{A}_{\text{i}}^{T} \mathbf{F}_{\text{pil}} \tag{2.28}$$

Note that $\mathbf{F}_{\text{pil}}$ is a force vector (acting at point **p** on body **i**) described in the local axes frame. Normally, force is conveniently described in the global axes. This equation can be rewritten as

$$\mathbf{f}_{\text{pi}} = \mathbf{A}_{\text{i}}^{T} \mathbf{T}_{\text{i}}^{T} \mathbf{F}_{\text{pi(3)}} \tag{2.29}$$

Now, $\mathbf{F}_{\mathrm{pi(3)}}$ is in the global axes frame of dimension 3; now premultiplying both sides of the equation with $\mathbf{T}'$

$$\mathbf{F}_{\mathrm{pi}} = \mathbf{T}'_{\mathrm{i}}\mathbf{A}_{\mathrm{i}}^{T}\mathbf{T}_{\mathrm{i}}^{T}\mathbf{F}_{\mathrm{pi(3)}} \tag{2.30}$$

The left-hand side of the equation has dimension 6. Where

$$\mathbf{T}'_{\mathrm{i}} = \begin{bmatrix} \mathbf{T}_{\mathrm{i}} & 0 \\ 0 & \mathbf{T}_{\mathrm{i}} \end{bmatrix} \tag{2.31}$$

Now, forces acting on body **i** are written as (for this, point deflections are described in the global axes)

$$\mathbf{F}_{\mathrm{pi(3)}} = \mathbf{k}_{\mathrm{p}}(\mathbf{T}_{\mathrm{j}}\mathbf{d}_{\mathrm{pj}} - \mathbf{T}_{\mathrm{i}}\mathbf{d}_{\mathrm{pi}}) \tag{2.32}$$

$$\mathbf{F}_{\mathrm{pi(3)}} = \mathbf{k}_{\mathrm{p}}(\mathbf{T}_{\mathrm{j}}\mathbf{A}_{\mathrm{pj}}\mathbf{u}_{\mathrm{j}} - \mathbf{T}_{\mathrm{i}}\mathbf{A}_{\mathrm{pi}}\mathbf{u}_{\mathrm{i}}) \tag{2.33}$$

u vectors are still in the local frame and need to be replaced by **U**.

$$\mathbf{F}_{\mathrm{pi(3)}} = \mathbf{k}_{\mathrm{p}}\left(\mathbf{T}_{\mathrm{j}}\mathbf{A}_{\mathrm{pj}}\mathbf{T}_{\mathrm{j}}'^{T}\mathbf{U}_{\mathrm{j}} - \mathbf{T}_{\mathrm{i}}\mathbf{A}_{\mathrm{pi}}\mathbf{T}_{\mathrm{i}}'^{T}\mathbf{U}_{\mathrm{i}}\right) \tag{2.34}$$

In this case,

$$\mathbf{T}'_{\mathrm{i}} = \begin{bmatrix} \mathbf{T}_{\mathrm{i}} & 0 \\ 0 & \mathbf{T}_{\mathrm{i}} \end{bmatrix} \tag{2.35}$$

$\mathbf{k}_{\mathrm{p}}$ is described in the global axes frame, $\mathbf{f}_{\mathrm{pi}}$ is a force vector in the global axes frame, and the following steps are self-explanatory.

$$\mathbf{M}_{\mathrm{i}}\ddot{\mathbf{u}}_{\mathrm{i}} = \sum_{\mathrm{p}}\mathbf{f}_{\mathrm{pi}} \tag{2.36}$$

$$\mathbf{M}_{\mathrm{i}}\mathbf{T}_{\mathrm{i}}'^{T}\ddot{\mathbf{U}} = \sum_{\mathrm{p}}\mathbf{f}_{\mathrm{pi}} \tag{2.37}$$

$$\mathbf{T}'_{\mathrm{i}}\mathbf{M}_{\mathrm{i}}\mathbf{T}_{\mathrm{i}}'^{T}\ddot{\mathbf{U}} = \mathbf{T}'_{\mathrm{i}}\sum_{\mathrm{p}}\mathbf{f}_{\mathrm{pi}} \tag{2.38}$$

$$\mathbf{T}'_{\mathrm{i}}\mathbf{M}_{\mathrm{i}}\mathbf{T}_{\mathrm{i}}'^{T}\ddot{\mathbf{U}} = \sum_{\mathrm{p}}\mathbf{F}_{\mathrm{pi}} \tag{2.39}$$

Substituting forces from Equation 2.34 into Equation 2.30 and into Equation 2.39 for suffix i and j, the final equation of motions may be written as

$$\mathbf{T}_\mathrm{i}'\mathbf{M}_\mathrm{i}\mathbf{T}_\mathrm{i}'^{T}\ddot{\mathbf{U}}_\mathrm{i}+\sum_\mathrm{p}\left(\mathbf{T}_\mathrm{i}'\mathbf{A}_\mathrm{i}^{T}\mathbf{T}_\mathrm{i}^{T}\mathbf{k}_\mathrm{p}\mathbf{T}_\mathrm{i}\mathbf{A}_\mathrm{pi}\mathbf{T}_\mathrm{i}'^{T}\right)\mathbf{U}_\mathrm{i}-\sum_\mathrm{p}\left(\mathbf{T}_\mathrm{i}'\mathbf{A}_\mathrm{i}^{T}\mathbf{T}_\mathrm{i}^{T}\mathbf{k}_\mathrm{p}\mathbf{T}_\mathrm{j}\mathbf{A}_\mathrm{pj}\mathbf{T}_\mathrm{j}'^{T}\right)\mathbf{U}_\mathrm{j}=0 \tag{2.40}$$

$$\mathbf{T}_\mathrm{j}'\mathbf{M}_\mathrm{j}\mathbf{T}_\mathrm{j}'^{T}\ddot{\mathbf{U}}_\mathrm{j}-\sum_\mathrm{p}\left(\mathbf{T}_\mathrm{j}'\mathbf{A}_\mathrm{j}^{T}\mathbf{T}_\mathrm{j}^{T}\mathbf{k}_\mathrm{p}\mathbf{T}_\mathrm{i}\mathbf{A}_\mathrm{pi}\mathbf{T}_\mathrm{i}'^{T}\right)\mathbf{U}_\mathrm{i}+\sum_\mathrm{p}\left(\mathbf{T}_\mathrm{j}'\mathbf{A}_\mathrm{j}^{T}\mathbf{T}_\mathrm{j}^{T}\mathbf{k}_\mathrm{p}\mathbf{T}_\mathrm{j}\mathbf{A}_\mathrm{pj}\mathbf{T}_\mathrm{j}'^{T}\right)\mathbf{U}_\mathrm{j}=0 \tag{2.41}$$

These are the final equations of motion described in matrix form ready to be assembled in the global matrix for the overall system.

It can be shown that using Equations 2.18 and 2.19, Equations 2.40 and 2.41 can be written in the following formats:

$$\begin{aligned}&\begin{bmatrix} m_\mathrm{i} & 0 \\ 0 & J_\mathrm{i}\end{bmatrix}\begin{Bmatrix}\ddot{x}_\mathrm{i}\\ \ddot{\theta}_\mathrm{i}\end{Bmatrix}-\begin{bmatrix} k_\mathrm{p} & k_\mathrm{p}R_\mathrm{pi} \\ R_\mathrm{pi}^{T}k_\mathrm{p} & R_\mathrm{pi}^{T}k_\mathrm{p}R_\mathrm{pi}\end{bmatrix}\begin{Bmatrix}x_\mathrm{i}\\ \theta_\mathrm{i}\end{Bmatrix}\\ &\quad+\begin{bmatrix} k_\mathrm{p} & k_\mathrm{p}R_\mathrm{pj} \\ R_\mathrm{pi}^{T}k_\mathrm{p} & R_\mathrm{pi}^{T}k_\mathrm{p}R_\mathrm{pj}\end{bmatrix}\begin{Bmatrix}x_\mathrm{j}\\ \theta_\mathrm{j}\end{Bmatrix}=\begin{Bmatrix}F_\mathrm{i}\\ M_\mathrm{i}\end{Bmatrix}\end{aligned} \tag{2.42}$$

$$\begin{aligned}&\begin{bmatrix} m_\mathrm{j} & 0 \\ 0 & J_\mathrm{j}\end{bmatrix}\begin{Bmatrix}\ddot{x}_\mathrm{j}\\ \ddot{\theta}_\mathrm{j}\end{Bmatrix}+\begin{bmatrix} k_\mathrm{p} & k_\mathrm{p}R_\mathrm{pi} \\ R_\mathrm{pi}^{T}k_\mathrm{p} & R_\mathrm{pi}^{T}k_\mathrm{p}R_\mathrm{pi}\end{bmatrix}\begin{Bmatrix}x_\mathrm{i}\\ \theta_\mathrm{i}\end{Bmatrix}\\ &\quad-\begin{bmatrix} k_\mathrm{p} & k_\mathrm{p}R_\mathrm{pj} \\ R_\mathrm{pj}^{T}k_\mathrm{p} & R_\mathrm{pj}^{T}k_\mathrm{p}R_\mathrm{pj}\end{bmatrix}\begin{Bmatrix}x_\mathrm{j}\\ \theta_\mathrm{j}\end{Bmatrix}=\begin{Bmatrix}F_\mathrm{j}\\ M_\mathrm{j}\end{Bmatrix}\end{aligned} \tag{2.43}$$

A more detailed treatment of this is given by Esat and Dabestani [1].

Overall equations of motion are now completed. It is worth restating that the stiffness and the damping matrices are identical in their structure. In order to obtain the damping matrix, all one needs to do is to replace the stiffness coefficients with corresponding damping coefficients.

2.3 Modeling Sensors and Actuators into the Model

An actuator located between q_j and q_j as shown in Figure 2.2 generates a force that is a function of the relative motion of end point attachments. More specifically, the force is a function of the relative motion along the line between these end points. In addition, the force is also controlled by feedback signals coming from sensors placed at different locations (k, l, m…). Signals corresponding to these positions d_k, d_l, d_m can be position, velocity, or acceleration or even a mixture of these and located on the same body as the attachment of the actuator or on different bodies.

2.3.1 Modeling Actuators

$$f = f((d_{qj} - d_{qi}), (d_k, d_l, d_m..)) \tag{2.44}$$

Relative motion of the end points of the actuator is given by

$$\dot{d}_q = (\dot{d}_{qj} - \dot{d}_{qi}) \tag{2.45}$$

And relative motion along the actuator may be expressed as

$$\dot{l}_q = \left(\frac{(R_{qj} - R_{qi})}{|R_{qj} - R_{qi}|} \right) \cdot (\dot{d}_{qj} - \dot{d}_{qi}) \tag{2.46}$$

And voltage due to this motion is given as

$$v_q = k\left(\frac{(R_{qj} - R_{qi})}{|R_{qj} - R_{qi}|} \right) \cdot (\dot{d}_{qj} - \dot{d}_{qi}) \tag{2.47}$$

Here, k is a magnetic field constant related to the actuator. Now, assuming the actuator is driven by external voltage v_e, the voltage balance equation can be written as

$$k\left(\frac{(R_{qj} - R_{qi})}{|R_{qj} - R_{qi}|} \right) \cdot (\dot{d}_{qj} - \dot{d}_{qi}) + L\frac{di}{dt} + Ri = v_e \tag{2.48}$$

The applied voltage may be treated in a number of different ways.

1. It can be treated as an external voltage supplied by a voltage source, entirely user controlled with no reference to the system state.
2. It can be a function of sensor signals, and these signals can be a function of state variables. And feedback signals may be due to absolute or relative body motions. Furthermore, these can be position, velocity, and displacements. The signals before being feedback can be processed to be filtered, magnified, differentiated, or integrated.
3. Measurement given by a sensor is along the sensor itself; therefore, additional transformation may be needed to relate to the state variables. This is not included in our treatment of sensor signals here although it is a relatively routine task.

As a general statement, the voltage as a function of sensor feedback may be written as

$$v_e = k_{sk}d_{sk} + k_{sl}d_{sl} + k_{sm}d_{sm} \tag{2.49}$$

Here, d signals can have different interpretations (depending on option 2 or 3).

To simplify the formulations and follow the simplified version presented in Equations 2.42 and 2.43, here it is assumed that the vectors are already in the global plane and additional spatial transformations will not be included.

First,

$$\dot{\mathbf{d}}_{qi} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{qi} & -y_{qi} \\ 0 & 1 & 0 & -z_{qi} & 0 & x_{qi} \\ 0 & 0 & 1 & y_{qi} & -x_{qi} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{oi} \\ \dot{y}_{oi} \\ \dot{z}_{oi} \\ \dot{\theta}_{xi} \\ \dot{\theta}_{yi} \\ \dot{\theta}_{zi} \end{Bmatrix} \text{ and}$$

$$\dot{\mathbf{d}}_{qj} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{qj} & -y_{qj} \\ 0 & 1 & 0 & -z_{qj} & 0 & x_{qj} \\ 0 & 0 & 1 & y_{qj} & -x_{qj} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{oj} \\ \dot{y}_{oj} \\ \dot{z}_{oj} \\ \dot{\theta}_{xj} \\ \dot{\theta}_{yj} \\ \dot{\theta}_{zj} \end{Bmatrix} \tag{2.50}$$

$$\dot{\mathbf{d}}_{qj} - \dot{\mathbf{d}}_{qi} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{qj} & -y_{qj} \\ 0 & 1 & 0 & -z_{qj} & 0 & x_{qj} \\ 0 & 0 & 1 & y_{qj} & -x_{qj} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{oj} \\ \dot{y}_{oj} \\ \dot{z}_{oj} \\ \dot{\theta}_{xj} \\ \dot{\theta}_{yj} \\ \dot{\theta}_{zj} \end{Bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & z_{qi} & -y_{qi} \\ 0 & 1 & 0 & -z_{qi} & 0 & x_{qi} \\ 0 & 0 & 1 & y_{qi} & -x_{qi} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{oi} \\ \dot{y}_{oi} \\ \dot{z}_{oi} \\ \dot{\theta}_{xi} \\ \dot{\theta}_{yi} \\ \dot{\theta}_{zi} \end{Bmatrix} \tag{2.51}$$

$$e_{\mathrm{q}} = \frac{(R_{\mathrm{qj}} - R_{\mathrm{qi}})}{\left|R_{\mathrm{qj}} - R_{\mathrm{qi}}\right|} \tag{2.52}$$

As these points are constant and known, it is better to replace Equation 2.52 by

$$e_{\mathrm{q}} = (e_{\mathrm{qx}}, e_{\mathrm{qy}}, e_{\mathrm{qz}}) \tag{2.53}$$

$$\begin{aligned} v_{\mathrm{q}} = k\left(\frac{(R_{\mathrm{qj}} - R_{\mathrm{qi}})}{\left|R_{\mathrm{qj}} - R_{\mathrm{qi}}\right|}\right)\cdot &\left\{ \begin{bmatrix} 1 & 0 & 0 & 0 & z_{\mathrm{qj}} & -y_{\mathrm{qj}} \\ 0 & 1 & 0 & -z_{\mathrm{qj}} & 0 & x_{\mathrm{qj}} \\ 0 & 0 & 1 & y_{\mathrm{qj}} & -x_{\mathrm{qj}} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{\mathrm{oj}} \\ \dot{y}_{\mathrm{oj}} \\ \dot{z}_{\mathrm{oj}} \\ \dot{\theta}_{\mathrm{xj}} \\ \dot{\theta}_{\mathrm{yj}} \\ \dot{\theta}_{\mathrm{zj}} \end{Bmatrix} \right. \\ &\left. - \begin{bmatrix} 1 & 0 & 0 & 0 & z_{\mathrm{qi}} & -y_{\mathrm{qi}} \\ 0 & 1 & 0 & -z_{\mathrm{qi}} & 0 & x_{\mathrm{qi}} \\ 0 & 0 & 1 & y_{\mathrm{qi}} & -x_{\mathrm{qi}} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{\mathrm{oi}} \\ \dot{y}_{\mathrm{oi}} \\ \dot{z}_{\mathrm{oi}} \\ \dot{\theta}_{\mathrm{xi}} \\ \dot{\theta}_{\mathrm{yi}} \\ \dot{\theta}_{\mathrm{zi}} \end{Bmatrix} \right\} \end{aligned} \tag{2.54}$$

$$\begin{aligned} v_{\mathrm{q}} = k(e_{\mathrm{qx}}, e_{\mathrm{qy}}, e_{\mathrm{qz}})\cdot &\left\{ \begin{bmatrix} 1 & 0 & 0 & 0 & z_{\mathrm{qj}} & -y_{\mathrm{qj}} \\ 0 & 1 & 0 & -z_{\mathrm{qj}} & 0 & x_{\mathrm{qj}} \\ 0 & 0 & 1 & y_{\mathrm{qj}} & -x_{\mathrm{qj}} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{\mathrm{oj}} \\ \dot{y}_{\mathrm{oj}} \\ \dot{z}_{\mathrm{oj}} \\ \dot{\theta}_{\mathrm{xj}} \\ \dot{\theta}_{\mathrm{yj}} \\ \dot{\theta}_{\mathrm{zj}} \end{Bmatrix} \right. \\ &\left. - \begin{bmatrix} 1 & 0 & 0 & 0 & z_{\mathrm{qi}} & -y_{\mathrm{qi}} \\ 0 & 1 & 0 & -z_{\mathrm{qi}} & 0 & x_{\mathrm{qi}} \\ 0 & 0 & 1 & y_{\mathrm{qi}} & -x_{\mathrm{qi}} & 0 \end{bmatrix} \begin{Bmatrix} \dot{x}_{\mathrm{oi}} \\ \dot{y}_{\mathrm{oi}} \\ \dot{z}_{\mathrm{oi}} \\ \dot{\theta}_{\mathrm{xi}} \\ \dot{\theta}_{\mathrm{yi}} \\ \dot{\theta}_{\mathrm{zi}} \end{Bmatrix} \right\} \end{aligned} \tag{2.55}$$

$$v_q = k\{e_{qx}, e_{qy}, e_{qz}\}\left\{\begin{bmatrix} 1 & 0 & 0 & 0 & z_{qj} & -y_{qj} \\ 0 & 1 & 0 & -z_{qj} & 0 & x_{qj} \\ 0 & 0 & 1 & y_{qj} & -x_{qj} & 0 \end{bmatrix}\begin{Bmatrix} \dot{x}_{oj} \\ \dot{y}_{oj} \\ \dot{z}_{oj} \\ \dot{\theta}_{xj} \\ \dot{\theta}_{yj} \\ \dot{\theta}_{zj} \end{Bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & z_{qi} & -y_{qi} \\ 0 & 1 & 0 & -z_{qi} & 0 & x_{qi} \\ 0 & 0 & 1 & y_{qi} & -x_{qi} & 0 \end{bmatrix}\begin{Bmatrix} \dot{x}_{oi} \\ \dot{y}_{oi} \\ \dot{z}_{oi} \\ \dot{\theta}_{xi} \\ \dot{\theta}_{yi} \\ \dot{\theta}_{zi} \end{Bmatrix}\right\} \tag{2.56}$$

It is easy to show that this now becomes

$$\begin{aligned} v_q = k\Big\{ & \left\{ e_{qx} \quad e_{qy} \quad e_{qz} \quad -e_{qy}z_{qj} + e_{qz}y_{qj} \quad e_{qx}z_{qj} - e_{qz}x_{qj} \quad -e_{qx}y_{qj} + e_{qy}x_{qj} \right\} \begin{Bmatrix} \dot{x}_{oi} \\ \dot{y}_{oj} \\ \dot{z}_{oj} \\ \dot{\theta}_{xj} \\ \dot{\theta}_{yj} \\ \dot{\theta}_{zj} \end{Bmatrix} \\ & - \left\{ e_{qx} \quad e_{qy} \quad e_{qz} \quad -e_{qy}z_{qi} + e_{qz}y_{qi} \quad e_{qx}z_{qi} - e_{qz}x_{qi} \quad -e_{qx}y_{qi} + e_{qy}x_{qi} \right\} \begin{Bmatrix} \dot{x}_{oi} \\ \dot{y}_{oi} \\ \dot{z}_{oi} \\ \dot{\theta}_{xi} \\ \dot{\theta}_{yi} \\ \dot{\theta}_{zi} \end{Bmatrix} \Big\} \end{aligned} \tag{2.57}$$

Elements of which are easy to locate in the analysis matrix (damping matrix).

2.3.2 Modeling Sensors and Feedback

Now consider a sensor located on mass ***k***. This is an arbitrary mass, and motion at the sensor position is given as

$$\mathbf{d}_{\text{sk}} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{\text{sk}} & -y_{\text{sk}} \\ 0 & 1 & 0 & -z_{\text{sk}} & 0 & x_{\text{sk}} \\ 0 & 0 & 1 & y_{\text{sk}} & -x_{\text{sk}} & 0 \end{bmatrix} \begin{Bmatrix} x_{\text{ok}} \\ y_{\text{ok}} \\ z_{\text{ok}} \\ \theta_{\text{xk}} \\ \theta_{\text{yk}} \\ \theta_{\text{zk}} \end{Bmatrix} \tag{2.58}$$

Note that here, if the sensor is a velocity sensor, then the displacement vector can be replaced by a velocity vector. Similarly for acceleration, the vector can be replaced by acceleration. Furthermore, for acceleration, in this representation, there is an implicit assumption that angular velocity terms are relatively small compared to acceleration, and thus second-order velocities contributing to accelerations are ignored. Assuming the feedback values of the sensors in x, y, z (global axes) are given as k_1, k_2, and k_3, then total feed, assuming they are all added, will be given as

$$v_{\text{e}} = \left\{ k_1 \quad k_2 \quad k_3 \quad -k_2 z_{\text{sk}} + k_3 y_{\text{sk}} \quad k_1 z_{\text{sk}} - k_3 x_{\text{sk}} \quad -k_1 y_{\text{sk}} + k_2 x_{\text{sk}} \right\} \begin{Bmatrix} x_{\text{ok}} \\ y_{\text{ok}} \\ z_{\text{ok}} \\ \theta_{\text{xk}} \\ \theta_{\text{yk}} \\ \theta_{\text{zk}} \end{Bmatrix} \tag{2.59}$$

However, if the sensor measures relative motion between mass $\boldsymbol{k}$ and mass $\boldsymbol{n}$, then

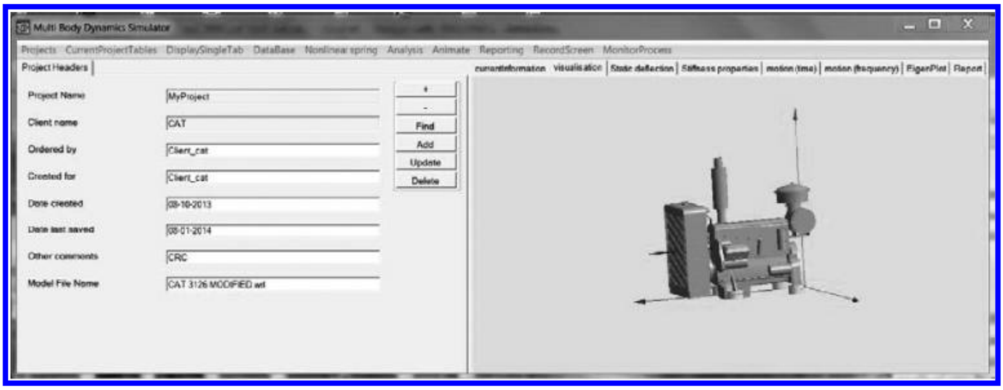
$$\begin{aligned} v_{\text{e}} = {} & \left\{ k_1 \quad k_2 \quad k_3 \quad -k_2 z_{\text{sn}} + k_3 y_{\text{sn}} \quad k_1 z_{\text{sn}} - k_3 x_{\text{sn}} \quad -k_1 y_{\text{sn}} + k_2 x_{\text{sn}} \right\} \begin{Bmatrix} x_{\text{on}} \\ y_{\text{on}} \\ z_{\text{on}} \\ \theta_{\text{xn}} \\ \theta_{\text{yn}} \\ \theta_{\text{zn}} \end{Bmatrix} \\ & - \left\{ k_1 \quad k_2 \quad k_3 \quad -k_2 z_{\text{sk}} + k_3 y_{\text{sk}} \quad k_1 z_{\text{sk}} - k_3 x_{\text{sk}} \quad -k_1 y_{\text{sk}} + k_2 x_{\text{sk}} \right\} \begin{Bmatrix} x_{\text{ok}} \\ y_{\text{ok}} \\ z_{\text{ok}} \\ \theta_{\text{xk}} \\ \theta_{\text{yk}} \\ \theta_{\text{zk}} \end{Bmatrix} \end{aligned} \tag{2.60}$$

To finalize the formulations, one needs to note that each sensor adds a new equation to the global matrix. Therefore, say for a three-mass system with a single actuator, the number of equations will be 19.

2.4 Introduction to Software MBDS

The formulations can only be tested by implementing them as software. Authors have previously developed a software package for modeling multiple bodies in 3-D space connected by spring and dashpots [1]. The software has been used in industry since the 1980s. The formulation given in this chapter is similar to those published previously with some additional generalization. Equations 2.42 and 2.43, however, are still more efficient for implementation purposes as some axes frame transformation can be performed before the assembly starts. The new version of VIBRATIO is called MBDS. An academic version of VIBRATIO is made available to readers of the *Vibration and Shock Handbook* [1] free of charge and was downloadable from www.signal-research.com. The company name has recently been changed to Dynamical Systems Technologies Ltd., and old VIBRTIO2002 is now downloadable from the new company website, www.dynasystech.co.uk.

The new software is far more sophisticated and an improved version of the old one with extensive additions. Apart from having all previous analysis capabilities, the new one has extensive graphical interfaces and solid graphics modeling capabilities, such as Boolean operations of solid objects and graphics import and export to many industrial CAD packages. The new software is completely Access database driven, and there is extensive use of Excel for manipulating nonlinear mounts as well as displaying output results. The Excel spreadsheets are also used for analysis session recording, which captures all the actions of the user. The last addition to the software is the control module. The full theory is given in this chapter. The software is at the prototyping stage. In other words, the actuator data and sensor data is not part of the software database system. After full testing, depending on the demand (the software MBDS is licensed by the Dynamical System Technologies Ltd.), the new module will fully integrated with the main software. The MBDS front screen is shown in Figure 2.3.


Multi Body Dynamics Simulator
Projects
CurrentProjectTables
DisplaySingleTab
DataBase
Nonlinear spring
Analysis
Animate
Reporting
RecordScreen
MonitorProcess
Project Headers
curantinformation
visualisation
Static deflection
Stiffness properties
motion (time)
motion (frequency)
FigerPlot
Report
+
-
Find
Add
Update
Delete

Project Name	MyProject
Client name	CAT
Ordered by	Client_cat
Created for	Client_cat
Date created	08-10-2013
Date last saved	08-01-2014
Other comments	CRC
Model File Name	CAT 3126 MODIFIED.wt

FIGURE 2.3
MBDS front screen.

2.4.1 A Simple Two-Mass Spring System with an Actuator and a Relative Velocity Sensor

A two-mass spring system is chosen as the test case. This is normally known as the quarter car model. The actuator is between two bodies acting in the middle as shown in Figure 2.4. In Figure 2.4a, the bottom mass is made invisible to show the location of the actuator. In Figure 2.4b, the bottom mass (mass 1) is made visible, showing the whole system. The sensor is not shown graphically as, when this chapter was submitted, no graphical object representing sensors had been decided. The sensor was a velocity sensor and located at the same location as the actuator measuring relative velocity between two connecting points.

2.4.2 Response of the System to a Simple Step Function

It is not practical to list all the data values relating to the problem. However, the geometry of the system can be seen from the diagram. A step force function is applied to mass number 1. In the first experiment, the actuator is not engaged. In the second experiment, the actuator is engaged. However, finding actuator feedback values and parameters have been by trial and error with no reference to a real actuator as the software does not have an actuator database. Fortunately, the trial and error process has been very quick, giving improvement in the responses, mostly because, in this example, the actuator acts no differently than a simple damper. This can easily be seen if the equation of the actuator and feedback is investigated. Off-center experiments would have been more interesting as one would demonstrate the effect of the actuator in modifying mode shapes and inducing modal coupling. Development of methods of optimization of the control scheme goes beyond the scope of this chapter. In Figure 2.5, results of a simple experiment are shown, and in Figure 2.5a, the system is analyzed under step force applied on mass 1, the bottom mass, without engaging the actuator. In Figure 2.5b, the actuator engaged with velocity feedback from the sensor. The relationship with feedback from sensors and the response initially improves the system behavior very quickly as feedback value has increased, but it soon settles with the behavior shown in Figure 2.5b. It is easy to see that the actuator

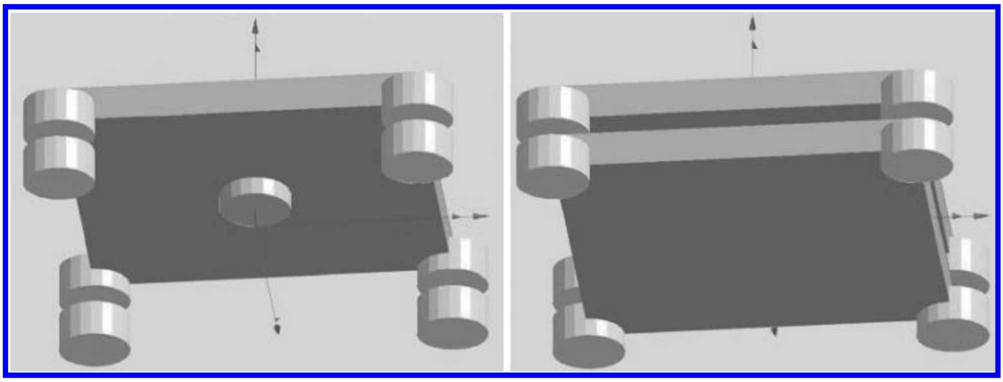
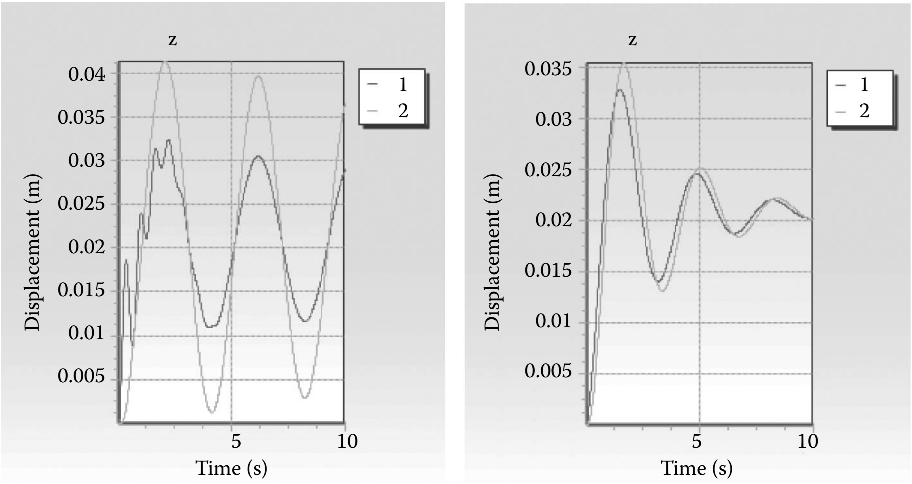

(a) (b)

FIGURE 2.4
Two-mass spring system with an actuator placed between two masses from a MBDS screen. (a) Actuator located in between the mass 1 and mass 2. (b) Combined system with the actuator concealed from the view.



(a) (b)

FIGURE 2.5
Time response of bodies subjected to a step function with no control and with control (from the MBDS screen). (a) Response with no control. (b) Controlled response.

is damping the motion between the two masses, but overall system damping is probably controlled by the ground mounts.

2.5 Conclusions

This chapter presented a method of analyzing the vibration and control of general multiple rigid body systems interconnected by linear springs and dashpots and controlled by multiple sensors and actuators. The mathematical modeling was developed for small vibrations with which nonlinear geometry effects and gyroscopic couplings are assumed to be small and negligible, and the deflection characteristics of mountings are assumed to be linear. There is a shortage of published material on general mathematical modeling of flexibly supported multiple rigid body systems for vibration analysis. No published material is available related to control of multiple rigid bodies as far as the authors are aware. This chapter gives precise and clear formulations suitable for computational implementation. Analysis options, although not discussed here, are the same as of VIBRATIO; MBDS can perform time domain vibration and control both for linear and materially nonlinear problems under a library of excitations as well as arbitrary excitations numerically generated or data logged from sensors with options for displacement, velocity, accelerations (as prescribed motion), or force. For a linear system, MBDS can perform Eigenvalue analysis as well as frequency analysis. In addition, MBDS can also perform analysis using excitations from power spectral density inputs as well as harmonic analysis, which is useful for investigating engine vibrations or any other system having multiple harmonics excitations. It is needless to say that all the analysis options listed here are for the system that now includes actuators driven by sensor feedback.

References

1. I. Esat and M. Dabestani, "Computer Analysis of Flexibly Supported Multibody Systems," *Vibration and Shock Handbook*, C. W. de Silva, Editor, Chapter 9, Taylor & Francis, CRC Press, Boca Raton, FL, 2005.
2. I. Esat and M. Dabestani, "Analysis of Flexible Multi Body Systems," *Computer Techniques in Vibration*, C. W. de Silva, Editor, Chapter 3, Taylor & Francis, CRC Press, Boca Raton, FL, 2007.
3. VIBRATIO, 2005. Information and download for the readers of this chapter, www.signal-research.com.
4. S. Timoshenko, D. H. Young, and W. Weaver, Jr., *Vibration Problems in Engineering*, 4th Edition, Wiley, New York, 1974.
5. J. P. Den Hartog, *Mechanical Vibration*, 4th Edition, McGraw-Hill, New York, 1956.
6. L. S. Jacobsen and R. S. Ayre, *Engineering Vibrations*, McGraw-Hill, New York, 1958.
7. R. F. D. Bishop and D. C. Johnson, *Mechanics of Vibration*, Cambridge University Press, New York, 1960.
8. L. Meirovitch, *Elements of Vibration Analysis*, 2nd Edition, McGraw-Hill, New York, 1986.
9. W. T. Thomson, *Theory of Vibrations with Applications*, 3rd Edition, Prentice-Hall, Englewood Cliffs, NJ, 1988.
10. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.
11. A. Ralston, *A First Course in Numerical Analysis*, McGraw-Hill, New York, 1965.
12. L. Meirovitch, *Analytical Methods in Vibrations*, Macmillan, New York, 1967.
13. T. K. Caughey, Classical Normal Modes in Damped Linear Dynamic Systems, *Journal of Applied Mechanics*, Vol. 27, 1960, pp. 269–271.

3

Mechanics of Materials

Yong Kang Chen

CONTENTS

3.1 Elastic Stress and Strain 38
  3.1.1 Introduction 38
  3.1.2 Load 39
  3.1.3 Stress 39
    3.1.3.1 Direct Stress: Tensile 39
    3.1.3.2 Direct Stress: Compressive 40
    3.1.3.3 Shear Stress 41
  3.1.4 Nonuniform Stress 41
  3.1.5 Complementary Shear Stresses 41
  3.1.6 Deformation 42
  3.1.7 Strain 42
    3.1.7.1 Direct Strain: Tensile 42
    3.1.7.2 Direct Strain: Compressive 43
    3.1.7.3 Shear Strain 43
    3.1.7.4 Volumetric Strain 43
  3.1.8 Elasticity and Yield 44
  3.1.9 Hooke's Law and Elastic Constants 44
3.2 Theory of Bending 45
  3.2.1 Introduction 45
  3.2.2 Definition 46
  3.2.3 Sign Convention of Bending Moment and Shearing Force 46
  3.2.4 Bending Moment and Shear Force Diagrams 47
    3.2.4.1 Freely Supported Beam with a Symmetrical Load 47
    3.2.4.2 Freely Supported Beam with an Asymmetrical Load 48
    3.2.4.3 Freely Supported Beam with Uniform Load w per Unit Run 50
  3.2.5 Bending Stresses 51
    3.2.5.1 Assumptions 51
    3.2.5.2 Beam Bending Strain 51
    3.2.5.3 Beam Bending Stress 52
    3.2.5.4 Section Properties 53
3.3 Deflection of Transverse Loaded Slender Beams 55
  3.3.1 Beam Deflection 55
  3.3.2 Flexure Equation 56
  3.3.3 Equilibrium and Determinacy 57
  3.3.4 Bending Moments 57
  3.3.5 Flexure Equation 58
  3.3.6 Deflection of a Transverse Loaded Beam 59

3.3.6.1 Cantilever with Point Load at Free End 59
3.3.6.2 Cantilever with Uniformly Distributed Load with Unit Length along the Whole Length 61
3.3.7 Deflection of Statically Indeterminate Beams 62
3.3.8 Beams with Discontinuous Bending Moment Equations 64
3.3.9 Singularity Function Method (Often Called Macaulay's Method) 65
3.4 Theory of Torsion 69
3.4.1 Introduction 69
3.4.2 Shear Strain/Stress Distribution 70
3.4.3 Torque *T* and Rate of Twist 70
3.4.4 Shear Stress from Torsion 72
3.5 Stress Transformation in Two Dimensions 72
3.5.1 Introduction 72
3.5.2 General State of Stress in Three Dimensions 72
3.5.3 General State of Stress in Two Dimensions 73
3.5.4 Analysis of Plane Stress in Two Dimensions 74
3.5.5 Calculation of Strains from Stresses 76
3.5.5.1 Hooke's Law 76
3.6 Strain Analysis and the Strain Gauge Rosettes 77
3.6.1 Introduction 77
3.6.2 Strain Gauge Rosettes 78
3.6.3 Conversion from Principal Strains to Principal Stresses 79
3.7 Mechanical Properties of Materials 80
3.7.1 Introduction 80
3.7.2 Tension and Compression Tests 80
3.7.3 Stress–Strain Behavior of Ductile Materials 80
3.7.4 Poisson's Ratio 82
3.8 Conclusions 83
References 83

SUMMARY The subject of mechanics is an important constituent of mechatronics. This chapter presents the fundamental theory of mechanics of materials, which can be employed for application in mechatronics. The chapter starts with three basic cases: axial loading, bending, and torsion. Then stress and strain are included, followed by analysis of displacements. Both stress and strain transformation are covered with strain gauge rosettes for measurement purposes in mechatronics.

3.1 Elastic Stress and Strain

3.1.1 Introduction

One of the foremost problems for a designer is to decide the proportions and shape of a machine or structure that will withstand the external forces imposed on it for an adequate working life without excessive wear, deformation, or fracture occurring. To satisfy these requirements necessitates the determination of the stresses and strains set up in the material. The concepts of stress and strain have been considered in one form or another for many hundreds of years, and in the past century and a half, mathematicians and engineers have developed the theories of elasticity. Although there are some formulae in existence that may be directly applicable to a designer's problem and only require numerical solution, there are many other problems for which there is no readily available answer. In those cases, the designer must be able to work from fundamental principles, perhaps making certain simplifying assumptions and drawing on past experience before a solution can be obtained.

This first section introduces some of the terminology and basic definitions, which then lead to a number of simple analyses of stress and strain in problems involving various forms of loading.

3.1.2 Load

The external forces acting on a piece of material constitute what is commonly called the "Load." The following are a few of the forces met with in practice:

1. A force due to a load not in motion—an example of which is a load hanging from a crane hook without being raised or lowered.
2. An inertia force due to change in velocity of a mass; this is met with in a piston within an engine practice and is the result of a change in velocity of the reciprocating parts.
3. A centrifugal force, which exists in pulleys and flywheels and is due to the tendency of a rotating mass to move away from its center of rotation.
4. A frictional force, resulting from the application of a brake on a drum in a car.
5. A force due to thermal expansion or contraction, which is met with in an aircraft fuselage, engines, and turbines.

Forces such as those mentioned above often act in such a manner as to cause tension, compression, shear, bending, or twisting in the material.

3.1.3 Stress

A piece of material in a static condition, subjected to external loads, will be in equilibrium under the action of those loads. However, if the material is to remain continuous in structure, then internal forces, which also must be in equilibrium, are set up to resist the external loads. The intensity of these internal forces is termed "stress" and is expressed as force per unit area in units such as N/mm^2, kN/mm^2, etc. The importance of stress is that it helps us to describe how two pieces of material, of different sizes, will react to the application of a particular value of external load. An excellent treatment of stress is found in textbooks [1–3].

Direct or normal stresses act in a direction perpendicular to a plane, and if outward from the plane, the stress is termed "tensile" and is distinguished by a positive sign. A stress toward a plane is called "compressive" and is negative in sign. Internal forces that act parallel to a plane are termed "shear stresses."

3.1.3.1 Direct Stress: Tensile

An example of a body stressed in this manner is shown in Figure 3.1, which represents a uniform vertical bar held at its upper end and carrying an axial load W. If the bar is cut at

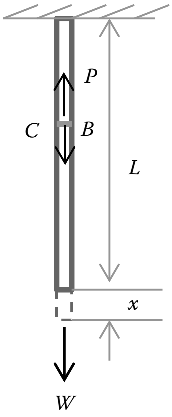


FIGURE 3.1
Tensile loading model.

C B, it is seen that a force *P*, acting in the opposite direction to *W* and equal in magnitude to *W*, is required to hold the lower portion in equilibrium. In the normal state of the bar, the force *P* is supplied by the internal forces in the material. If the section *C B* is taken at any point between the ends of the bar, a similar condition exists (neglecting the weight of the portion of the bar below the section).

The intensity of tensile stress at *C B* is given by

$$\sigma = \frac{W}{A} \tag{3.1}$$

where *A* is the cross-sectional area of the bar.

3.1.3.2 Direct Stress: Compressive

Figure 3.2 represents a uniform vertical bar carrying a load *W*, which, in this case, causes a stress opposite in nature to that discussed in tensile stress. The line of action is again assumed to be axial.

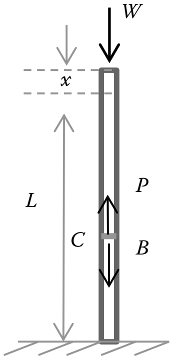


FIGURE 3.2
Compression loading model.

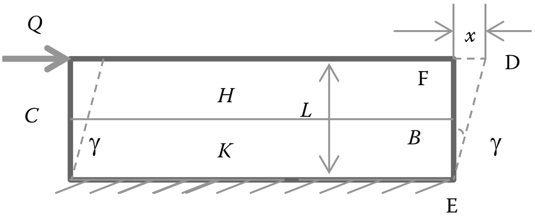


FIGURE 3.3
Typical shear model.

Considering a section $C B$, it is seen that a force P, equal in magnitude to W and acting in the opposite direction to W, is required for equilibrium. Again, neglecting the weight of the portion of the bar above $C B$, the intensity of the compression stress at $C B$ is given by

$$\sigma = -\frac{W}{A} \tag{3.2}$$

where A is the cross-sectional area of the bar.

3.1.3.3 Shear Stress

Stress of this nature is said to occur on a section of a body if, on opposite faces of the section, equal and opposite parallel forces exist. Let a rectangular block of metal be soldered to a heavy mass of iron, and suppose a force Q is to be applied, parallel to a face of cross-sectional area A as shown in Figure 3.3.

Now consider the section $C B$. The upper portion H exerts a shear force Q on the face of the lower section K, which, in turn, exerts an equal and opposite force on the face of the upper portion H. The intensity of the shear stress on the section at $C B$ is given by

$$\tau = \frac{Q}{A} \tag{3.3}$$

3.1.4 Nonuniform Stress

In each of the cases considered above, the force is assumed to be distributed uniformly over the surface. If this not the case, then the intensity of stress at a point on the surface is taken to be equal to the limiting ratio of $\delta W/\delta A$ or $\delta Q/\delta A$ when each is reduced indefinitely, where δW or δQ is the force acting on the very small area δA.

3.1.5 Complementary Shear Stresses

In Figure 3.4, let OABC be a rectangular block of unit thickness. Suppose a shear stress equal in magnitude to τ_1 acts on the faces CB and OA, these being taken perpendicular to the plane of the paper. The forces caused by these stresses form a couple, and for equilibrium of the block, they must be balanced by another couple. Let τ_2 be the magnitude of the intensity of the forces forming the balancing couple and acting on the faces OC and AB. Then τ_2 has the faces OC and AB in a state of shear.

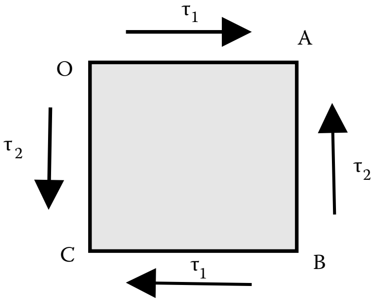


FIGURE 3.4
Pure shear stress element.

Taking moments about O for equilibrium of the block, we have

$$\tau_2 \times \text{area of face AB} \times \text{OA} = \tau_1 \times \text{area of face BC} \times \text{OC}$$

or

$$\tau_2 \times a \times 1 \times b = \tau_1 \times b \times 1 \times a$$

Therefore,

$$\tau_2 = \tau_1$$

Thus, it is seen that a shear stress is automatically accompanied by a shear of equal intensity but opposite turning moment in a direction perpendicular to that of the original shear stress. These are termed complementary shear stresses.

3.1.6 Deformation

No material is rigid, and therefore, the application of external loads to a material or component causes deformation. This may be quite large and visible if the material is, say, rubber; on the other hand, the same load applied to a piece of steel of similar size to the rubber would require a sensitive instrument to measure the deformation. A study of deformation is an essential part of the mechanics of solids as shown in Section 3.3.6.

3.1.7 Strain

Strain is a nondimensional measure of the deformation produced by internal or external forces.

3.1.7.1 Direct Strain: Tensile

In Figure 3.1, it can be observed that the deformation is an elongation of the amount x, and if L is the initial length of the bar, then the tensile strain is given by

$$\varepsilon = \frac{x}{L} \tag{3.4}$$

or is the elongation per unit of length and is taken to be positive.

3.1.7.2 Direct Strain: Compressive

In Figure 3.2, the deformation is shown to be a shortening of the bar by the amount x, and if L is the unloaded length, the compressive strain is given by

$$\varepsilon = -\frac{x}{L} \tag{3.5}$$

or the shortening per unit of length and is taken to be negative.

3.1.7.3 Shear Strain

The state of deformation produced by shear is shown by the dotted lines in Figure 3.3. The movement x of the corner is exceedingly small, so EFD may be regarded as a right-angled triangle. The shear strain is given by

$$\gamma \approx \tan\gamma = \frac{x}{L} \tag{3.6}$$

because the angle γ is very small.

3.1.7.4 Volumetric Strain

Another kind of strain is encountered if a solid, such as a cube, is subjected to forces on each face. Figure 3.5 shows a cube subjected to equal compressive normal forces on each face. (This state of loading would be obtained if the cube was lowered to a great depth in a liquid.) The result of the forces on each face will be a shortening of each edge of the cube and hence a reduction in volume. The cubical or volumetric strain is given by

$$\varepsilon = \frac{\text{change in volume}}{\text{original volume}} \tag{3.7}$$

or change in volume per unit of volume.

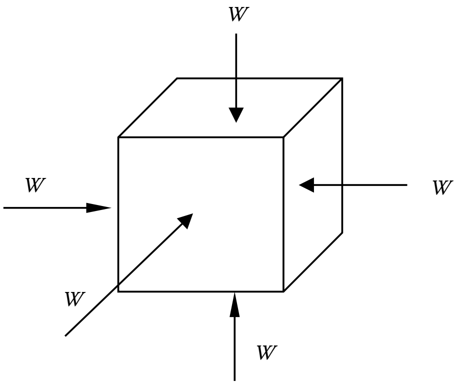


FIGURE 3.5
A 3-D stress element under equal compressive loading.

If l is the length of the edge of the unstrained cube and x the decrease in length of each edge, then the change in volume is given by $3l^2x - 3lx^2 + x^2$. Now because x is very small, terms containing it or higher powers may be neglected, and hence the change in volume is approximately $3l^2x$. The volumetric strain is therefore approximately $3x/l$, which is three times the linear strain of the edge.

It is important to note that, in each case, the strain is a ratio of displacement and length and is thus nondimensional.

Let the dimensions of a bar be x, y, and z, and suppose the bar is loaded in such a manner that the corresponding strains in the direction of the above dimensions are ε_x, ε_y and ε_z. Any of these strains may be positive or negative, depending on whether they are elongations or contractions.

The new dimensions of the bar are $x(1 + \varepsilon_x)$, $y(1 + \varepsilon_y)$ and $z(1 + \varepsilon_z)$.

Volume of the unstrained bar = xyz

$$\begin{aligned}\text{Volume of the strained bar} &= x(1+\varepsilon_x)\,y(1+\varepsilon_y)\,z(1+\varepsilon_z)\\ &= xyz(1+\varepsilon_x+\varepsilon_y+\varepsilon_z)\end{aligned} \tag{3.8}$$

The product of two or more strains has been neglected as the strains are very small quantities, so the volume strain can be given as follows:

$$\varepsilon_v = \frac{\text{change in volume}}{\text{original volume}} = \varepsilon_x + \varepsilon_y + \varepsilon_z \tag{3.9}$$

3.1.8 Elasticity and Yield

A material is said to be perfectly elastic if the strain due to loading disappears with the removal of the load and also if the strain for a given value of load during the unloading process is equal to the strain for the same value of load during the loading process.

A limiting value of load will be found at which the strain does not completely disappear with the removal of the load. The value of the stress corresponding to this load is called the elastic limit, and the residual strain is referred to as a permanent set. Beyond the elastic limit, the material enters the plastic range and yielding occurs. From the yield point onward, nonrecoverable deformation is caused by increasing the load. (See Section 3.7 for a detailed discussion on this behavior.)

3.1.9 Hooke's Law and Elastic Constants

It was discovered by Hooke (in 1678) that if a material is loaded within the elastic range, then the deformation produced is proportional to the load applied. Now stress and strain are proportional to load and deformation, respectively, and therefore stress is linearly related to strain. Hence,

$$\frac{\text{stress}}{\text{strain}} = \frac{W}{A} \times \frac{L}{x} = k\frac{W}{x} \tag{3.10}$$

where $k = L/A$, and because x is proportional to W, then $k(w/x)$ is equal to a constant; that is, stress = a constant × strain.

This constant is called the modulus of elasticity, and its magnitude will depend on the material and on the nature of the stress and strain dealt with. Because stress is a force per unit of area and strain is nondimensional, the modulus of elasticity will have units of force per unit area.

When a body is subjected to simple tension or compression, the modulus of elasticity is usually called Young's modulus, and it is invariably denoted by the letter E. Shear stress and shear strain are also linearly related in the elastic range of the material and the constant of proportionality is termed the shear or rigidity modulus and is generally denoted by the letter G (sometimes C). Volumetric strain is proportional to the hydrostatic stress indicated in Figure 3.5, and the constant relating them is termed the bulk or volume modulus denoted by K. These constants are related to one another.

Using the appropriate symbols for stress and strain, the elastic constants may be written as

$$\text{Direct stress:}\ \ E = \frac{\sigma}{\varepsilon} \tag{3.11}$$

$$\text{Shear stress:}\ \ G = \frac{\tau}{\gamma} \tag{3.12}$$

$$\text{Hydrostatic stress:}\ \ K = \frac{\sigma_v}{\varepsilon_v} \tag{3.13}$$

where σ_v denotes the intensity of the normal stress on each face of the cube in Figure 3.5.

The discussion above and in previous paragraphs indicates a method of attack in problems involving the mechanics of solids:

1. A condition of equilibrium can be expressed in terms of the applied external forces and the internal reacting forces, that is, the stress acting on a certain area of the body.
2. The strain is expressed in terms of a simple assumed geometry of deformation.
3. There are the *stress–strain relationships* in terms of the elastic constants.

There are thus three basic equations: stress related to applied force strain to deformation and stress to strain formation with which the unknowns can be determined. The present and succeeding sections are devoted to the application of the above approach to various problems.

3.2 Theory of Bending

3.2.1 Introduction

The last section was devoted to the problems involving direct stress of uniform intensity over the cross-section of a component. When a bar supported in some manner is subjected to lateral loading, the form of deformation is termed "bending." Both direct and shear stresses are set up, neither of which is uniform over the section of the bar. Many engineering components in service are subjected to forces that cause bending. Both the bending stresses and deflection in such cases are concerned.

3.2.2 Definition

Consider the horizontal beam as shown in Figure 3.6, which is at equilibrium under the forces R_1 and R_2 acting vertically upward and vertical downward forces W_1, W_2, and W_3. The forces keeping the portion to the right-hand side of a section at A in equilibrium are $R_2 - W_3$, and those exerted by the left-hand portion of the beam across the section A are $R_1 - W_1 - W_2$.

As show in Figure 3.6, a thin longitudinal slice of the beam as at A, containing point A, is kept in equilibrium by two opposite parallel forces that must be equal in magnitude. Thus, the slice is in a state of shear. The shearing force at any point along a loaded beam is the algebraic sum of all the vertical forces acting to one side of the point.

The forces R_2 and W_3 have a resultant moment acting on the thin slice in a counterclockwise direction, and forces R_1, W_1, and W_2 have a resultant moment acting on the slice in a clockwise direction. Because the slice in the beam has no rotary motion, the two resultant moments must be equal in magnitude. The bending moment at any point along a loaded beam is the algebraic sum of the moments of all the vertical forces acting to one side of the point.

3.2.3 Sign Convention of Bending Moment and Shearing Force

It is necessary to express the value of the bending moment and shearing force at any point along the span. These are usually expressed as a step function across the span in which a particular function is expressed in a region between two loading positions. It is customary to adapt a sign convention for the bending moment and shearing force functions. A force tending to bend a beam so that its center of curvature is above the beam, as shown in Figure 3.7, is said to produce positive bending or sagging, and a force that tends to shear the left-hand portion of the beam upward with respect to the right-hand portion results in a shear.

When a force tends to cause a beam to be bent with the center of curvature below the beam, it is said to produce negative bending or hogging as shown in Figure 3.7. Negative shear is caused when the left-hand portion of the beam tends to shear downward with respect to the right-hand portion. Distances x measured along the beam are taken as positive when measured from the left to right.

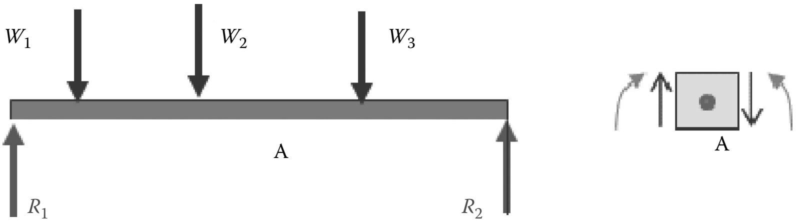


FIGURE 3.6
Planar bending model.

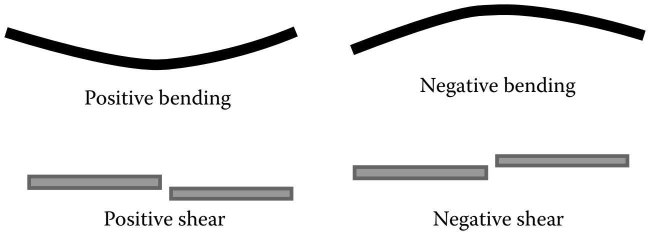


FIGURE 3.7
Sign convention.

3.2.4 Bending Moment and Shear Force Diagrams

It is necessary to express the value of the bending moment and shearing force at any point along the span. These are usually expressed as a step function across the span in which a particular function is expressed in a region between two loading positions.

3.2.4.1 Freely Supported Beam with a Symmetrical Load

As shown in Figure 3.8, from symmetry, the reaction at each support is $P/2$. Considering the left-hand half of the span, the bending moment at any point S, in the region of AB, a distance x from the left-hand support, is given by

$$M_S = \frac{P}{2}x \tag{3.14}$$

The bending moment, therefore, gradually increases from zero at the support, where x is zero, to $\frac{1}{2}P \times \frac{L}{2} = \frac{1}{4}PL$, where x is $\frac{L}{2}$. Equation 3.14 is also a function of position of x, the position of point S.

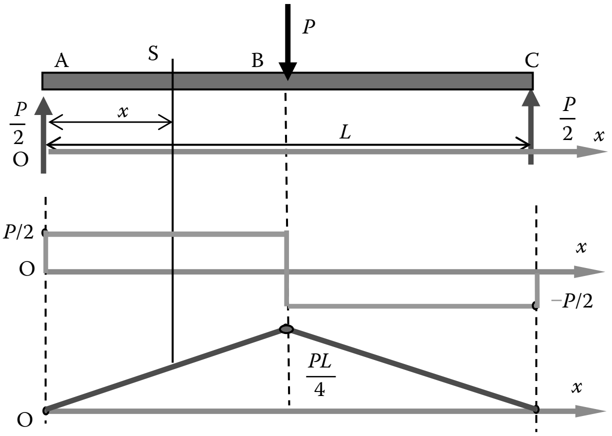


FIGURE 3.8
Shear force and bending moment diagrams due to a central load point.

Considering the second loading region of BC, the bending moment at any point S, in the region of BC, a distance x from the left-hand support, is given by

$$M_S = \frac{P}{2}x - P\left(x - \frac{L}{2}\right) \tag{3.15}$$

From bending moment function of Equation 3.15, it can be seen clearly that the bending moment takes its maximum value when $x = L/2$, the midspan of the beam as follows:

$$M_{\max} = \frac{PL}{4} \tag{3.16}$$

3.2.4.2 Freely Supported Beam with an Asymmetrical Load

As shown in Figure 3.9, let the load divide the span into two loading regions, AB and BC. The length of each region is a and b, respectively. Because the load is not at midspan, the reaction at each end will be different. Taking moments about C, we have

$$R_1L = Pb$$

therefore,

$$R_1 = \frac{Pb}{L}$$

and

$$R_1 + R_2 = P$$

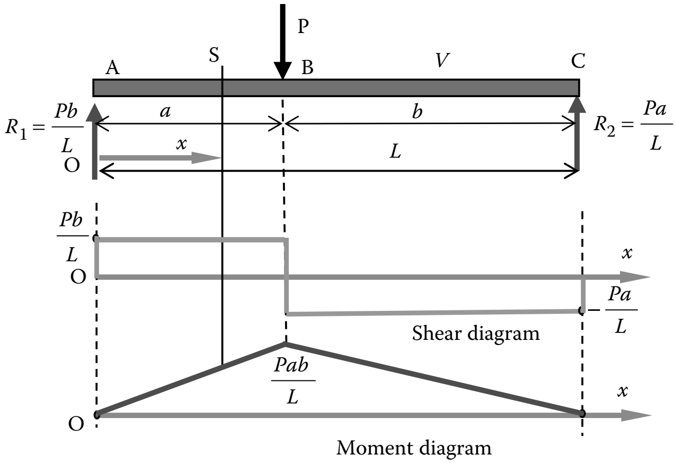

P
A
S
B
V
C
$R_1 = \frac{Pb}{L}$
a
b
$R_2 = \frac{Pa}{L}$
x
L
O
$\frac{Pb}{L}$
O
x
$\frac{Pa}{L}$
Shear diagram
$\frac{Pab}{L}$
x
O
Moment diagram

FIGURE 3.9
Shear force and bending moment diagrams due to a noncentral load point.

therefore,

$$\begin{aligned} R_2 &= P - R_1 \\ &= P - \frac{Pb}{L} \\ &= P\left(\frac{L-b}{L}\right) = \frac{Pa}{L} \end{aligned}$$

Considering the left-hand half of the span, the bending moment at any point S, in the region of AB, a distance x from the left-hand support, is given by

$$M_S = \frac{Pb}{L}x \tag{3.17}$$

Equation 3.17 is also a function of the position of x, the position of point S, and called a bending moment function. The bending moment function represents the bending moment distribution in the first loading region AB. Based upon the bending moment function, the bending moment diagram in region AB will be triangular, increasing from zero at $x = 0$ to (Pb/L_a) when $x = a$.

The above bending moment function cannot apply for the second loading region of BC because we have to cross the point of the concentrated load P, which represents a discontinuity in a mathematical function. In region BC, we have to express a new bending moment function to reflect the bending moment distribution in it. The new function can be obtained as follows:

Considering the second loading region of BC, the bending moment at any point V, in the region of BC, a distance x from the left-hand support, is given by

$$M_V = \frac{Pb}{L}x - P(x-a) \tag{3.18}$$

This gives a triangular bending moment diagram in region BC, the bending moment decreasing from Pab/L where x is a, to

$$M_V = \frac{Pb}{L}L - P(L-a) = 0$$

where $x = L$. The maximum bending moment is thus given by

$$M_{\max} = \frac{Pab}{L} \tag{3.19}$$

The shearing force Q over the left-hand portion of the first loading region is uniform, and

$$Q = \frac{Pb}{L}$$

After passing P, the shearing force becomes

$$Q = \frac{Pb}{L} - P = -\frac{Pa}{L}$$

and is uniform over the right-hand portion of the span, the second loading region.

3.2.4.3 Freely Supported Beam with Uniform Load w *per Unit Run*

As shown in Figure 3.10, the total load on the beam is wL, and from symmetry, the reaction at each end is $\frac{wL}{2}$. As the load crosses the whole span, the bending moment function can be expressed as

$$M_A = \frac{wL}{2}x - wx\frac{x}{2} \tag{3.20}$$

This is because the loading is continuing. The bending moment diagram is parabolic in form, increasing from zero, where x is zero, to a maximum value

$$M_{\max} = \frac{wL}{2}\frac{L}{2} - w\frac{L^2}{8} = \frac{wL^2}{8} = \frac{WL}{8}$$

where $x = \frac{L}{2}$ at midspan and $W = wL$.

The total shear load is given by $M_A = \frac{wL}{2} - wx$.

The shearing force diagram is thus triangular, decreasing linearly from $\frac{wL}{2}$ when $x = 0$ to zero when $x = L/2$. There is also a triangular diagram for the right-hand side of the beam,

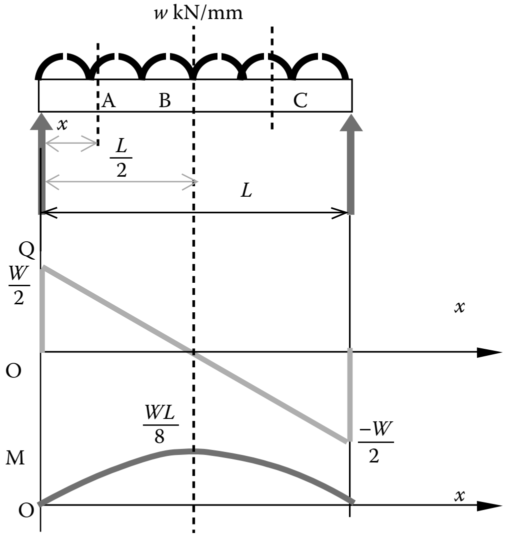


FIGURE 3.10
Shear force and bending moment diagrams due to a uniform load.

the shearing force increasing from zero when $x = L/2$, to $Q = \frac{wL}{2}$ when $x = L$. Therefore, $Q_{\max} = \frac{wL}{2} = \frac{W}{2}$ at each end of the beam.

3.2.5 Bending Stresses

3.2.5.1 *Assumptions*

When a beam is bent due to the application of a constant bending moment, that is, by couples to both its ends without being subjected to shear, it is said to be in a state of pure bending. The following assumptions are employed to obtain the bending stresses:

1. Young's modulus has the same value for the materials of the beam in tension as in compression, and the stress is proportional to the strain.
2. A transverse section of the beam, which is a plane before bending, will remain a plane after bending.
3. The radius of curvature of the beam before bending is very large in comparison with the transverse dimensions of the beam.
4. The resultant pull or thrust across a transverse section of the beam is zero.
5. The transverse section of the beam is symmetrical about an axis, passing through the centroid of the section and parallel to the plane of the beam.
6. There is no lateral stress acting on the beam.

3.2.5.2 *Beam Bending Strain*

To understand the bending stress in an arbitrarily loaded beam, consider a small element cut from the beam as shown in the Figure 3.11. The beam type or actual load does not affect the derivation of the bending strain equation. Details are found in textbooks [1–3]. Recall that the basic definition of normal strain is

$$\varepsilon = \Delta L/L. \tag{3.21}$$

In Figure 3.11, using the line segment, AB, the before and after length can be used to give

$$\varepsilon = \frac{G'H' - GH}{GH} \tag{3.22}$$

The line length GH is the same for all locations before bending. However, the length $G'H'$ becomes shorter above the neutral axis (for positive moment) and longer below. The line GH and $G'H'$ can be described using the radius of curvature, ρ, and the differential angle, $d\theta$.

$$G'H' = \rho d\theta$$

$$G'H' = (\rho - y)d\theta$$

Notice that the y coordinate is assumed upward from the neutral axis where there is no strain. Putting these together gives

$$\varepsilon = \frac{(\rho - y)d\theta - \rho d\theta}{\rho d\theta} = \frac{\rho d\theta - y d\theta - \rho d\theta}{\rho d\theta} = -\frac{y}{\rho} \tag{3.23}$$

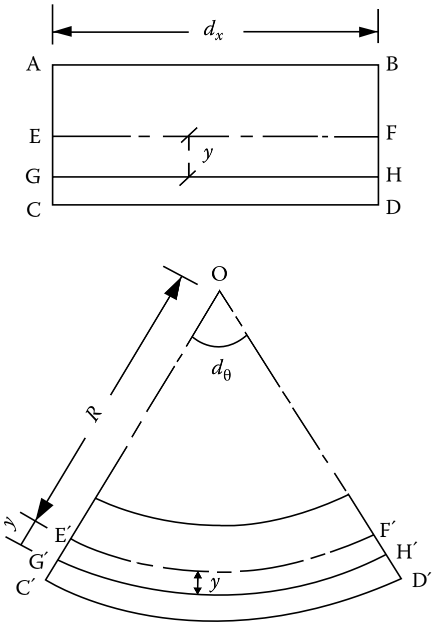


FIGURE 3.11
A beam before and after bending.

This relationship gives the bending strain at any location as a function of the beam curvature and the distance from the neutral axis. However, this equation is of little use, and needs to be converted to stress. Also, the radius of curvature is difficult to determine at a given beam location.

3.2.5.3 Beam Bending Stress

The strain equation above can be converted to stress by using Hooke's law $\sigma = E\varepsilon$, giving

$$\sigma = -\frac{Ey}{\rho} \tag{3.24}$$

The relationship can be determined by summing the moment due to the normal stresses on an arbitrary beam cross-section and equating it to the applied internal moment. This is the same as applying the moment equilibrium equation about the neutral axis (NA).

$$\begin{gathered}\sum M_{\text{NA}} = 0,\\ \int y(-dF) = M,\\ -\int y\sigma\, dA = M\end{gathered} \tag{3.25}$$

For a positive moment, the top stresses will be in compression (negative stress), and the bottom stresses will be in tension (positive stress) and thus the negative sign in the equation. This equation can be changed by using Equation 3.24:

$$\frac{E}{\rho} = \int y^2\, dA = M \tag{3.26}$$

It is interesting to note that the integral is the area moment of inertia, *I*, or the second moment of the area. Many handbooks list the moment of inertia of common shapes. A review of moment of inertia is given below in the next subsection. Using the area moment of inertia gives

$$\frac{EI}{\rho} = M \tag{3.27}$$

But the radius of curvature, ρ, is still there. Equation 3.24, $\rho = -Ey/\sigma$, can be used again to eliminate ρ, giving

$$\frac{EI}{\frac{-Ey}{\sigma}} = M \tag{3.28}$$

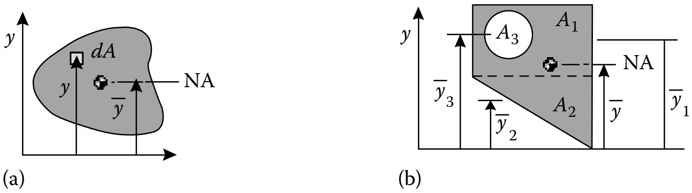
Simplifying and rearranging gives

$$\sigma = -\frac{My}{I} \tag{3.29}$$

This equation gives the bending normal stress and is also commonly called the flexure formula. The y term is the distance from the neutral axis (up is positive). The I term is the moment of inertia about the neutral axis.

3.2.5.4 Section Properties

Section property in bending includes the position of centroid and second moment of area. As shown in Equation 3.29, bending stress is a function of y, a position away from the neutral axis, and I, the second moment of area. Bending stress takes its maximum value when y takes its maximum value, $\bar{y}$, the location of centroid. Both $\bar{y}$ and I need to be determined before bending stress can be calculated using Figure 3.12.



(a)

(b)

FIGURE 3.12
Centroid of a section. (a) Centroid for arbitrary shape. (b) Centroid based on subshapes.

3.2.5.4.1 *Location of the Neutral Axis* $\bar{y}$

If the cross-section is symmetrical about the horizontal axis, then the neutral axis is halfway between the top and bottom. However, for a nonsymmetrical beam, such as a "T" cross-section, the neutral axis is not halfway between the top and bottom and needs to be determined before the bending stress equation can be used.

The neutral axis is located at the centroid (geometric center) of the cross-section. Recall from statics, the centroid can be found using two methods. The first is by integration:

$$\bar{y} = \frac{\int y\,dA}{\int dA} \tag{3.30}$$

The second and more common method is the method of parts. The beam cross-section is split into geometric shapes that are common (rectangle, triangle, circle, etc.). The centroid of basic shapes can be found in handbooks, eliminating the need for integration. The centroid is

$$\bar{y} = \frac{\sum \bar{y}_i dA_i}{\sum dA_i} \tag{3.31}$$

If there is a hole, then that area is considered to be negative, and the same equation can still be used. As an example, the diagram at the right hand end in Figure 3.13 would be,

$$\bar{y} = \frac{\bar{y}_1 A_1 + \bar{y}_2 A_2 + \bar{y}_3 A_3}{A_1 + A_2 + A_3} \tag{3.32}$$

3.2.5.4.2 *Second Moment of Area (Area Moment of Inertia) I*

As shown in Figure 3.13, similar to the centroid, the area moment of inertia can be found by either integration or by parts. The moment of inertia is also called the "second moment of the area" because that describes the integration equation

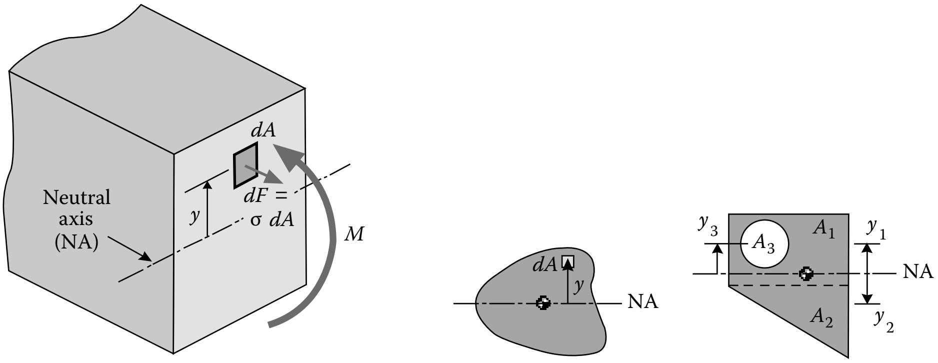


FIGURE 3.13
Area moment of inertia determination.

$$I = \int y^2 \, dA \tag{3.33}$$

When using this with the bending stress equation, I is about the neutral axis and not the x-axis.

A more common method to find the moment of inertia is by parts. Like finding the centroid (needs to be done first), the object is split into smaller basic shapes. The moment of inertia about the centroid of each part can be found in a handbook. Then the individual moments of inertia are moved to the neutral axis using the parallel axis theorem. For a particular subshape, this gives

$$I_{\mathrm{NA}-i} = I_i + y_i^2 A_i \tag{3.34}$$

where I_i is the moment of inertia about its own shape, and $I_{\mathrm{NA}-i}$ is the moment of inertia about the object's neutral axis.

All the moment of inertia terms can then be added together to give

$$I_{\mathrm{NA}} = \sum I_{\mathrm{NA}-i} \tag{3.35}$$

For the diagram at the right-hand end in Figure 3.13, the parts method gives

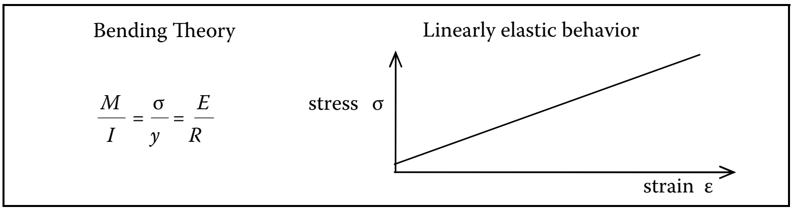
$$I = \left(I_1 + y_1^2 A_1\right) + \left(I_2 + y_2^2 A_2\right) + \left(I_3 + y_3^2 A_3\right) \tag{3.36}$$

Notice, for a hole, the moment of inertia is subtracted for that shape.

3.3 Deflection of Transverse Loaded Slender Beams

3.3.1 Beam Deflection

Section 3.2.5 developed methods for finding internal stresses in beams due to the bending moments caused by applied loads. The theory of bending (TB) relates bending stress to bending moment. In its simplest form, it helps us to "stress" beams of symmetrical cross-section made from materials with linearly elastic stress/strain characteristics.


Bending Theory
$\frac{M}{I} = \frac{\sigma}{y} = \frac{E}{R}$
Linearly elastic behavior
stress σ
strain ε

In passing, we should note that the shear forces also present in the beam will cause shear stress which is *not* predicted by TB. Other theories are available to deal with shear.

In many practical applications, the amount by which a beam deforms is as important as its ability to support loads without becoming overstressed. In machine design, control of deflections is necessary to maintain operating clearances. Damaging vibrations can be the result of excessive deflections in parts of aircraft structure. Upper limits are placed on deflections in building frameworks to avoid poor appearance, cracking, and flexibility of the structure.

It is necessary to develop methods to calculate transverse deflections of the whole beam. Such deflections are due to the bending moment and to the shear forces, but it is possible to separate the two effects. This section of the present chapter deals with deflections due to bending only.

3.3.2 Flexure Equation

In Figure 3.14, θ is the angle that the tangent to the curve at A makes with the x-axis, and $\theta - d\theta$ is that which the tangent at B makes with the same axis. The normal to the curve at A and B meet at O. The point O is the center of curvature, and R is the radius of curvature of the small portion AB of the deflection curve of the neutral axis.

$$\frac{1}{R} = \frac{d\theta}{ds} \tag{3.37}$$

Numerically, using the sign convention in Section 3.2.3, it can be seen that a positive increase in ds is accompanied by a negative value of $d\theta$. Thus, when signs are taken into account, the right-hand side of the above equation becomes negative.

Deflections of the neutral axis are denoted by the symbol v, measured positive downwards, and are relatively small giving a flat form of deflection curve; therefore, no error is introduced in assuming that

$$ds \approx dx \quad \theta \approx \tan\theta = \frac{dv}{dx} \quad \text{hence} \quad \frac{d\theta}{ds} = \frac{d\theta}{dx} = \frac{d\left(\frac{dv}{dx}\right)}{dx} = \frac{d^2v}{dx^2}$$

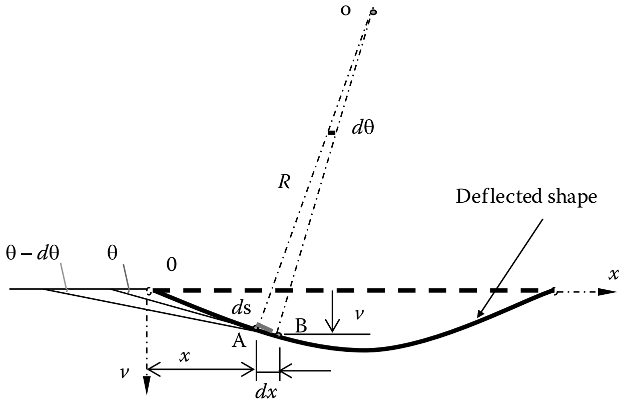


FIGURE 3.14
Curvature of the deflected beam.

$$\frac{1}{R} = -\frac{d\theta}{ds} = -\frac{d^2v}{dx^2}$$
$$\frac{EI}{R} = M \tag{3.38}$$

When elastic bending occurs

$$EI\left(-\frac{d^2v}{dx^2}\right) = M \tag{3.39}$$

This is the differential equation of the deflection curve; v has been deliberately chosen to be positive downward, so that with the beam deformed as shown, moment M, deflection v, and distances from the neutral axis y are each positive and consistent with each other. This equation is called the flexure equation and portrays the relationship between flexure of the beam axis and an applied load.

3.3.3 Equilibrium and Determinacy

We will be analyzing beams that have transverse loads in the $\boldsymbol{y}$ direction and moments such as $\boldsymbol{M}$ about $\boldsymbol{z}$ axes as shown in Figure 3.15. The normal six equations of static equilibrium will reduce down to *two* because there are no forces in the $\boldsymbol{x}$ and $\boldsymbol{z}$ directions and no moments about the $\boldsymbol{x}$ and $\boldsymbol{y}$ axes. This leaves us with $\Sigma F_y = 0$ and $\Sigma M_z = 0$.

With *two* equations of equilibrium, we can solve for *only two* unknowns. When a real beam is idealized, there are two possibilities:

1. There are *two* reactions, for example, a cantilever or a simply supported beam (Figure 3.15), and *two* reactions can be found using the *two* equations of equilibrium, for example, a force equation and a moment equation.
2. There are *three or more* reactions (Figure 3.16). Extra reactions are called redundancies. Three reactions equal one redundancy, etc.

3.3.4 Bending Moments

Loads applied to beams cause internal stresses to arise, and they can be separated into bending stresses. These can be analyzed as stress or their effects can be described as

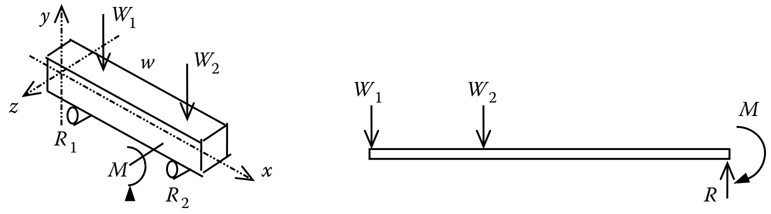


FIGURE 3.15
Beam with two point loads.

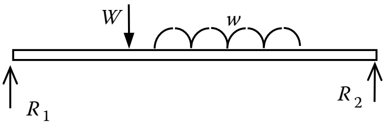


FIGURE 3.16
Beam and simple supports with both a point load and a uniform load.

bending moment and *shear force* carried by the beam at any perpendicular cross-section. In turn, the bending moment and shear force can be quantified by considering the *equilibrium* of part of the beam with the external loads and reactions supported by that part. For example, to find the shear force and bending moment at a cross-section AA on the beam shown in Figure 3.17, take one part—left hand in the calculation, but right hand would do equally well—and perform an equilibrium calculation to find F and M.

$$\sum F_y = 0 \quad \text{i.e.,}\ W - R_1 - F = 0. \quad \text{Hence}\ F = W - R_1$$

$$\sum M_{\text{AA}} = 0 \quad \text{i.e.,}\ R_1 b - W(b-a) - M = 0. \quad \text{Hence}\ M = R_1 b - W(b-a)$$

This is to say that the ending moment at any cross-section is numerically equal to the algebraic sum of all external moments due to the applied loads and reactions on one side of the chosen cross-section. It does not matter which side is used but both sides cannot be included simultaneously in the expression because the algebraic sum will be *zero* if such a mistake is made. Because it is the algebraic sum, a sign convention for positive and negative bending moments is needed as described in Section 3.2.3.

3.3.5 Flexure Equation

This is the mathematical basis for the work on beam deflections and on struts. From $\frac{M}{I} = \frac{E}{R}$, it can be noted that M is linked with R (radius of curvature). Zero M gives infinite R or a straight beam.

Finite M gives finite R, that is, curvature and, hence, deflection. So we already have a theory that implicitly links bending moment with transverse deflection [3].

With some substitution, we get a second-order differential equation. The flexure equation of Equation 3.39 can be expressed as

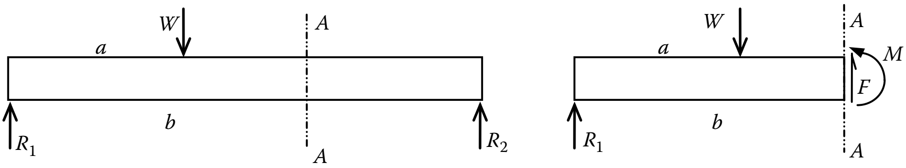


FIGURE 3.17
Beam and simple supports with point load W.

$$EI\frac{d^2v}{dx^2} = -M \tag{3.40}$$

Rearrange it and integrate it with respect to x:

$$\frac{dv}{dx} = \text{slope} = \int -\frac{M}{EI}dx \tag{3.41}$$

Integrate it again:

$$v = \text{deflection} = \iint -\frac{M}{EI}dx.dx \tag{3.42}$$

To make use of normal (nongraphical, non-numerical) methods of integration, the bending moment must be expressible as a function of *x*, the position on the beam.

When the BM can be expressed as a function of *x* that is relatively easy to integrate, the determination of *dv*/*dx* and *v* is simple. The most readily integrated functions of *x* are those that are continuous from one end of the beam to the other. That is, *one* equation gives the BM at *all* points on the beam. It should be clear that this is only going to be true for a limited number of straightforward loading cases. (Even a simply supported beam with on-point load needs two equations to fully describe the change of BM along it; think of the shape of the BM diagram.) So here are a few simple cases to help get into the method.

3.3.6 Deflection of a Transverse Loaded Beam

3.3.6.1 Cantilever with Point Load at Free End

The built-in end is equivalent to two reactions: a force *R* and a moment *M*. It also prevents vertical and rotational movement at 2. The real beam will take up a shape something like Figure 3.18 (not to scale).

Figure 3.19 shows a hogging BM and will be described as negative. The bending moment is a single straight line, so one equation covers the whole length of the beam.

Put $x = 0$ at the LH end.

Then BM at position *x* is given by $M = -Wx$

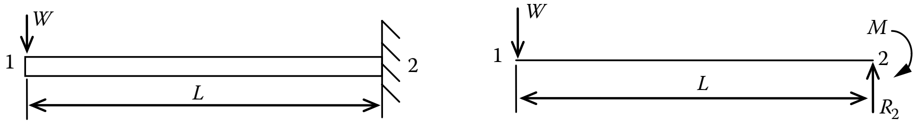
This equation gives the value of *M* for any position *x* on the beam between 0 × <*L*. It is a continuous equation.

Notice that, in this case, there was no need to calculate R_2 and M_2. Now we use the flexure Equation 3.39.

$$EI\frac{d^2v}{dx^2} = -M = -(-Wx) = Wx \tag{3.43}$$

Integrate with respect to x:

$$EI\frac{dv}{dx} = \frac{Wx^2}{2} + A \tag{3.44}$$



Real cantilever

Idealized cantilever

FIGURE 3.18
Cantilever beam with a point force of *W*.

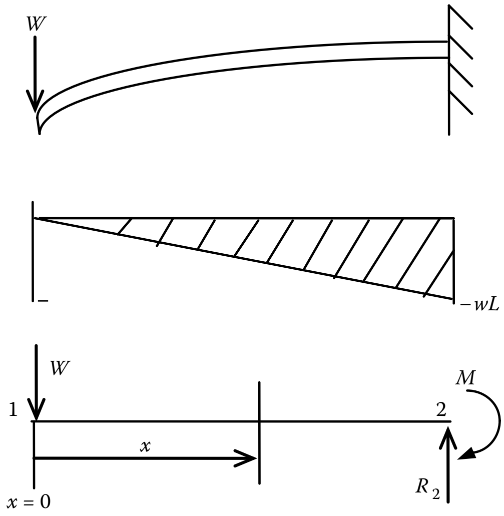


FIGURE 3.19
Bending moment diagram of a cantilever beam.

and again:

$$EIv = \frac{Wx^3}{6} + Ax + B \tag{3.45}$$

Next step: Find the constants *A* and *B* from the *boundary conditions.*

Two known facts about slope and/or deflections at $x = L$, slope = 0 and deflection = 0 are needed. Therefore,

$$\text{at } x = L, \frac{dv}{dx} = 0 \text{ substitute into Equation 3.5} \quad 0 = \frac{WL^2}{2} + A \quad A = -\frac{WL^2}{2}$$

$$\text{at } x = L, \text{v} = 0 \quad \text{substitute, along with value of } A \text{ into Equation 3.6}$$

$$0 = \frac{WL^3}{6} - \frac{WL^2}{2} \cdot L + B \quad B = \frac{WL^3}{3}$$

So rewriting Equations 3.43 and 3.44 gives

$$\text{slope} \,\ldots\ldots\, \frac{dv}{dx} = \frac{1}{EI}\left[\frac{Wx^2}{2} - \frac{WL^2}{2}\right] \tag{3.46}$$

$$\text{deflection .. } v = \frac{1}{EI}\left[\frac{Wx^3}{6} - \frac{WL^2x}{2} + \frac{WL^3}{3}\right] \tag{3.47}$$

The problem is now solved because these equations can be used to find values of slope and deflection at any position along the beam by inserting the appropriate values of x.

Checking free end ($x = 0$):

$$\text{slope ....} \quad \frac{dv}{dx_{\text{free end}}} = -\frac{WL^2}{2EI} \quad \text{and deflection}\ldots v_{\text{free end}} = +\frac{WL^3}{3EI} \tag{3.48}$$

This has illustrated the method in its simplest form. We will deal with more complicated loading cases and will have to write more involved BM equations. Also, for discontinuous equations, there will be another technique, but this case has covered the central points.

3.3.6.2 Cantilever with Uniformly Distributed Load with Unit Length along the Whole Length

As shown in Figure 3.20, the beam is idealized, and support conditions are $x = L$, $dv/dx = 0$, and $v = 0$.

Equation for BM at x

$$M = -\frac{Wx^2}{2}$$

The term $-wx^2/2$ for the BM due to u.d. load is found by working out how much load in the length x, that is, wx and multiplying this force by the distance from the cross-section back to the center of gravity of the load contained on length x about the position defined by x. So the flexure equation becomes

$$EI\frac{d^2v}{dx^2} = -M = \frac{wx^2}{2} \tag{3.49}$$

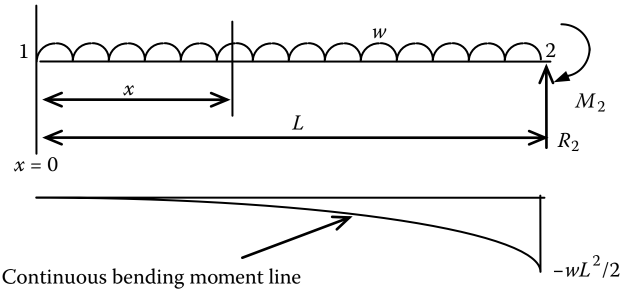


FIGURE 3.20
Bending moment diagram with a uniform load.

Integrating

$$EI\frac{dv}{dx} = \frac{wx^3}{6} + A \tag{3.50}$$

Again

$$EIv = \frac{wx^4}{24} + Ax + B \tag{3.51}$$

from the *boundary conditions* at $x = L$, slope = 0, and deflection = 0.

Therefore,

$$\text{at } x = L,\ \frac{dv}{dx} = 0 \text{ substitute into Equation 3.50} \quad 0 = \frac{wL^3}{6} + A \quad A = -\frac{wL^3}{6}$$

$$\text{at } x = L, v = 0 \text{ substitute, along with value of } A \text{ into Equation 3.51}$$

$$0 = \frac{wL^4}{24} - \frac{wL^3}{6}\cdot L + B \quad B = \frac{wL^4}{8}$$

The job is now done. Both slopes and deflections at any points on the cantilever can be found as, for example, at the free end ($x = 0$):

$$\text{slope} \ldots\ldots \frac{dv}{dx}_{\text{free end}} = -\frac{wL^3}{6EI} \quad \text{and deflection} \ldots v_{\text{free end}} = +\frac{wL^4}{8EI} \tag{3.52}$$

3.3.7 Deflection of Statically Indeterminate Beams

As previously agreed, if there are more than two reactions, they cannot be found from the two equations of static equilibrium. But the flexure equation may be of use now. The beam shown in Figure 3.21 is built in at the right-hand end and simply supported at the left-hand end so that all the deflection there is prevented.

There are three unknown reactions: R_1, R_2, and M_2, but we have an *extra known fact*: $v = 0$ at end 1. At position x,

$$EI\frac{d^2v}{dx^2} = -M = -R_1x + \frac{wx^2}{2} \tag{3.53}$$

$$EI\frac{dv}{dx} = -R_1\frac{x^2}{2} + \frac{wx^3}{6} + A \tag{3.54}$$

$$EIv = -R_1\frac{x^3}{6} + \frac{wx^4}{24} + Ax + B \tag{3.55}$$

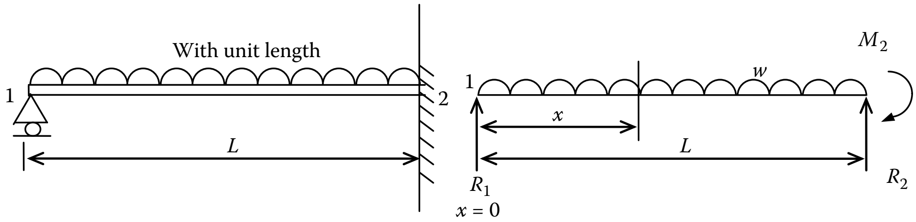


FIGURE 3.21
Beam configuration.

Known data:

$$x = 0, v = 0$$

$$x = L, v = 0$$

$$x = L, dv/dx = 0$$

Make three substitutions of the known data to get the two constants, A and B, and also the reaction R_1.

$$x = 0, v = 0 \text{ into Equation 3.55 produces } B = 0$$

$$x = L, v = 0 \text{ into Equation 3.55 gives } 0 = -R_1\frac{L^3}{6} + \frac{wL^4}{24} + Ax \tag{3.56}$$

$$x = L, dv/dx = 0 \text{ into Equation 3.54 gives } 0 = -R_1\frac{L^2}{2} + \frac{wL^3}{6} + A \tag{3.57}$$

Equations 3.56 and 3.57 are simultaneous and solve to give

$$R_1 = \frac{3wL}{8} \quad A = \frac{wL^3}{48}$$

With R_1, the indeterminacy is broken. We could now use the two equations of equilibrium to find the remaining two reactions.

$$\sum F_y = 0 \quad R_1 + R_2 = wL \quad R_2 = \frac{5wL}{8}$$

$$\sum M_2 = 0 \quad M_2 + R_1L - wL\cdot\frac{L}{2} = 0 \quad M_2 = \frac{wL^2}{8}$$

If we had been wanting to know about deflections, slopes, etc., then we could have used the values R_1, A, and B in Equations 3.54 and 3.55 to get the required answers, for example, slope at propped end 1 from Equation 3.41 with $x = 0$

$$\frac{dv}{dx}_{\text{propped end 1}} = \frac{C_1}{EI} = \frac{wL^3}{EI}$$

Deflection at midspan can be determined from Equation 3.55 with $x = L/2$

$$v_{\text{mid span}} = \frac{1}{EI}\left[\frac{-3wL}{8}\cdot\frac{L^3}{3}\cdot\frac{1}{6} + \frac{wL^3}{24}\cdot\frac{L^4}{16} + \frac{wL^3}{48}\cdot\frac{L}{2}\right] = \frac{2wL^4}{384EI} = \frac{wL^4}{192EI}$$

3.3.8 Beams with Discontinuous Bending Moment Equations

So far, we have solved problems in which one function of x is enough to describe the variation of bending moment along the entire length of the beam. This stage will deal with beams in which the bending moment equations are discontinuous, and a modified form of integration will be introduced. However, the basic method will not change, so it is important that you have a good grasp of the main steps. We do not have to look very far for the first example: a simply supported beam with a single point load, shown idealized in Figure 3.22 together with its bending moment diagram.

It is apparent from the shape of the *BM* diagram (two different straight lines) that we need two separate *BM* equations to describe the two parts of the *BM* diagram.

Equation A	$BM = R_1x$	Valid	$0 \le x \le a$
Equation B	$BM = R_1x - W(x - a)$	Valid	$a \le x \le L$

The change from one equation to the other at x = a is called a *discontinuity*, that is, the bending moment is a discontinuous function of x. Although we might expect that this will introduce some added complications, it does not change the underlying method. We still use the flexure equation, and our task is still to integrate that equation.

One approach would be to integrate the two equations separately, but this would generate *four* constants of integration and require *four* pieces of known data in order to proceed to a solution. And that is with only one discontinuity. Although possible, this approach rapidly becomes unmanageable. There is a need for an easier approach to use in these cases of discontinuities.

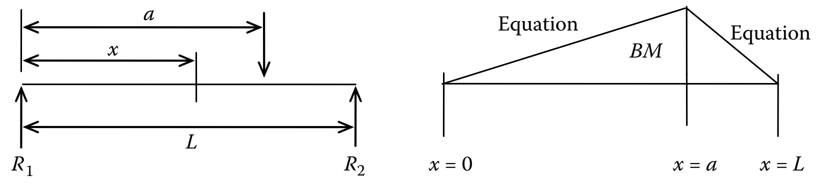


FIGURE 3.22
A beam with a point load.

3.3.9 Singularity Function Method (Often Called Macaulay's Method)

The obvious labor in using the long method to solve discontinuous cases can be diminished by making use of singularity functions. Before going into detail, look again at the two equations for the point-loaded beam in Figure 3.9.

Equation A	$BM = R_1x$	Valid	$0 \le x \le a$
Equation B	$BM = R_1x - W(x - a)$	Valid	$a \le x \le L$

Notice that Equation A is the first term of Equation B, that is, Equation B contains Equation A.

Figure A's second point load is brought in, and hence, a second discontinuity is shown in Figure 3.23.

The equations for the bending moment in the three parts of the beam are the following:

Equation A	$BM = R_1x$	Valid	$0 \le x \le a$
Equation B	$BM = R_1x - W_1(x - a)$	Valid	$a \le x \le b$
Equation C	$BM = R_1x - W_1(x - a) - W_2(x - b)$	Valid	$b \le x \le L$

Now, Equation C contains both Equations B and A. In other words, we can reduce Equation C to B or A by "dropping off" one term at a time whenever we need to.

This suggests the basis of a method for dealing with this type of problem with a discontinuous bending moment function. We write down one bending moment equation for the part of the beam furthest away from the origin of x. This equation will "contain" all the others. We integrate this *one* equation only to give *one* slope equation and *one* deflection equation with only *two* constants of integration. Then, whenever we need to, we "drop off" terms from the BM, slope, or deflection equations to make them applicable to any other part of the beam.

The problem is how to drop off terms. Look at Equations A, B, and C in the case of Figure 3.23. Equation B is relevant to the middle part of the beam, and the term $-W_4(x - b)$ has to be dropped from Equation C to produce Equation B.

What would happen if we put into the term $-W_2(x - b)$? *Try it!*

Put $x = b$ into $-W_2(x - b)$	This gives *zero*
Put $x < b$ into $-W_2(x - b)$	This gives a *negative* result inside the bracket.

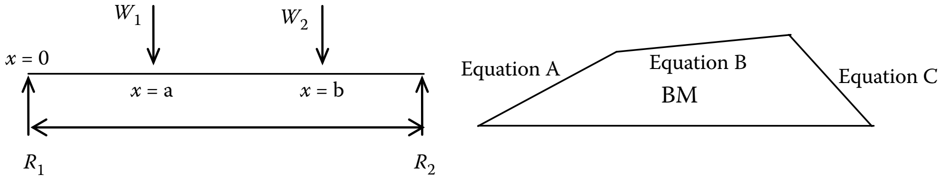


FIGURE 3.23
Modified beam.

The zero looks after itself, but the negative result is the one that is interesting to us because it gives us a rule by which to work these problems. And it was this that Macaulay spotted.

If a value of x that defines a position or region on a beam is substituted into the *single* BM equation for the beam and produces a negative result for the contents of any of the bracketed terms in the equation, then those terms are not relevant to that position or region of the beam and must be "dropped out," that is, there is no such thing as a negative value inside these brackets.

Functions of x that behave in this way are called singularity functions.

Check again on the previous case: put $x <$ a into Equation C.

$$\text{Bending Moment} = R_1x - W_1(x - a) - W_2(x - b)$$

This time, the second and third terms would produce negative results inside the brackets. So they must be dropped off the equation leaving only BM = R_1x which is correct because $x <$ a defines the left segment of the beam.

It is obvious that we need to pay particular attention to these brackets throughout the calculations. So we will adopt special bracket signs to distinguish them thus [1,2]. This is meant to remind us to apply the test

Are the contents positive or negative?

A more "mathematical" way of expressing all this is by defining special singularity functions with defined properties.

$$\text{for example, } [x - x_0]^n \quad n \text{ is an integer including } 0.$$

x_0 is the value of x at the initial boundary of the beam to which the function applies.

The singularity function takes the following values—this is just another way of expressing what we have already done above.

$$\begin{aligned}[x - x_0]^n &= [x - x_0]^n \quad \text{when } n > 0 \text{ and } x > x_0 \\ &= 0 \quad \text{when } n > 0 \text{ and } x < x_0\end{aligned}$$

$$[x - x_0]^0 = 1 \text{ when } x > x_0$$

$$[x - x_0]^0 = 0 \text{ when } x < x_0$$

Integration of singularity (Macaulay) functions

$$\int [x - x_0]^n \, dx = \frac{1}{n+1}[x - x_0]^{n+1} + \text{constant} \tag{3.58}$$

Integration of these functions must be done in this way. It is essential to the method that we keep these bracketed terms $[x - x_0]$ *intact* throughout the integration. If you do not keep the brackets intact, you cannot apply the "drop out" test when it is needed.

Let's go back to the single point-loaded beam and solve it using this new technique. Simply supported beam, length L, with point load W at distance from LH end (Figure 3.24).

Put $x = 0$ at LH end. Select position x in RH part of beam, that is, $x > a$ farthest away from $x = 0$, and write down the BM equation at x.

$$\text{Bending moment} = R_1x - W[x - a]$$

Put this into the flexure equation

$$EI\frac{d^2v}{dx^2} = -M = -R_1x + W[x - a]$$

Note, in a numerical question, we would first calculate R_1 from the equilibrium of the whole beam.

Also W would have a value.

Now integrate

$$EI\frac{dv}{dx} = -R_1\frac{x^2}{2} + W\frac{[x-a]^2}{2} + A \tag{3.59}$$

and again

$$EI \cdot v = -R_1\frac{x^3}{6} + W\frac{[x-a]^3}{6} + Ax + B \tag{3.60}$$

now substitute the known data $x = 0$, $v = 0$ and $x = L$, $v = 0$ into Equation 3.60 and at the same time employ the "drop out" test.

$$x = 0, v = 0 \quad EI.(0) = -\frac{R_1}{6}(0)^3 + \frac{W}{6}[0-a]^3 + A(0) + B$$

$$(0) = -\,0 + \text{drops out} + 0 + B$$

The second term on the right-hand side must "drop out" because the contents of the [ ] bracket in the above equation would be negative.

The result is $B = 0$

Now put $x = L$, $v = 0$ into Equation 3.60.

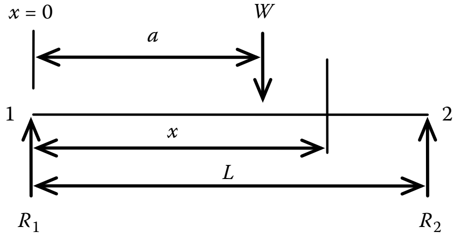


FIGURE 3.24
Beam with point load.

$$EI.(0) = -R_1 \frac{L^3}{6} + W \frac{[L-a]^3}{6} + AL$$

The second term has now stayed in because the contents $[L - a]$ of the bracket are now positive. The equation will solve to give A, and the problem is virtually done.

But remember to apply the "drop out" test all the way through when you are inserting known data and when you want to find slope and deflection at chosen positions on the beam.

For example, suppose we want to find deflection v at midspan $x = L/2$.

As the beam is drawn, $L/2$ is $<a$, so when we put $x = L/2$ into the bracket $[x - a]$, it becomes $[L/2 - a]$, which would be *negative*, so the term must be dropped.

This would give the equation: at $x = \mathrm{L}/2$ $EI \cdot v = -R_1 \frac{(L/2)^3}{6} + A(L/2)$.

We will do a numerical problem later, but first there are other types of load to consider, together with their singularity functions for use in the BM equations.

The discontinuity considered has been a "kink" in the BM diagram due to the presence of a point load. Only two other types need to be detailed: discontinuities due to the beginning and end of a distributed load and ones due to localized applied moments.

Remember that a discontinuity occurs when either an extra term appears in the BM equation or a term drops out. This is the same as saying that a change has taken place in the shape of the BM diagram. In our problems there are only three basic loads: point forces u.d. loads and localized moments. These give three basic shapes in the BM diagram: straight lines, curved lines, and vertical steps.

So, we could look at a BM diagram and deduce how many discontinuities are present and where they are. For example, the BM diagram shown in Figure 3.25 has *three* changes of shape.

It starts with a straight line. There is a step at point 1. A curve starts at point 2. It changes to a straight line at point 3. (We could deduce the loading.)

We will need type functions $[x - x_0]^n$ to account for the discontinuities. These are the functions to be included in the single BM equation for a beam segment furthest from $x = 0$, as shown in Figure 3.26.

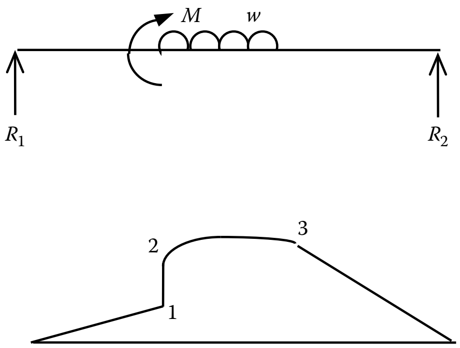
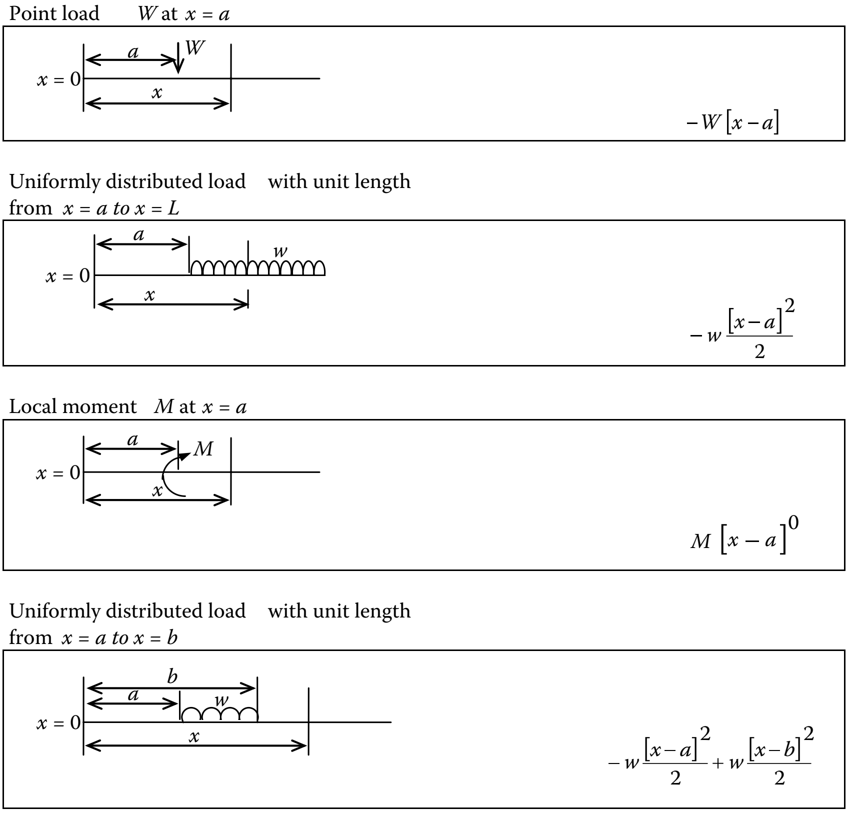


FIGURE 3.25
BM diagram.



Point load W at $x = a$

Uniformly distributed load with unit length from $x = a$ to $x = L$

Local moment M at $x = a$

Uniformly distributed load with unit length from $x = a$ to $x = b$

FIGURE 3.26
BM diagram determination.

3.4 Theory of Torsion

3.4.1 Introduction

Shear stress and strain have been defined in Sections 3.1.3 through 3.1.7. An important problem involving shearing is that in which material is subjected to deformation by twisting. A shaft is a typical component subjected to twisting in both power and motion transmission. It is necessary to have a torsion theory to calculate shear stress and shear deformation.

In order to calculate the torsional stresses in the shaft with a circular section, it is assumed that

1. The shaft is straight and of uniform cross-section over its length.
2. Torque is constant and transmitted along the bar by each section trying to shear over its neighbor.
3. Transverse planes remain parallel to each other.
4. For a small angle of rotation, the length of the shaft and its radius remain unchanged.
5. The material must be *homogeneous*.

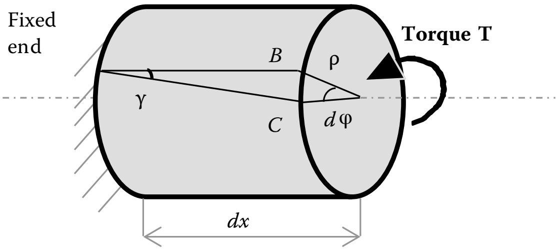


FIGURE 3.27
Twist due to torsion.

3.4.2 Shear Strain/Stress Distribution

Figure 3.27 shows a small differential element from the shaft. Assume the left end is fixed and the right end rotates by $d\varphi$ due to the applied torque T, where $d\varphi$ is termed the *angle of twist* of the element. The surface of radius ρ rotates through angle γ, which is shear strain. The arc is defined as length BC, which is equal to

$$da = \rho d\varphi = \gamma dx$$

Rearranging the above equation gives

$$\gamma = \rho \frac{d\varphi}{dx} \tag{3.61}$$

$\frac{d\varphi}{dx}$ expresses the rate of twist and is constant for the cross-sectional plane. Equation 3.61 states that the magnitude of shear strain for any of these elements varies only with its radial distance ρ.

By using Hooke's law,

$$\tau = G\gamma \tag{3.62}$$

and substituting for shear strain γ, Equation 3.61, Equation 3.62 becomes that

$$\tau = \rho G \frac{d\varphi}{dx} \tag{3.63}$$

Equation 3.63 relates the shear stress linearly to the distance ρ away from the center of the section. As a result, the shear stress distribution then looks as Figure 3.28. A complete treatment of stress, strain, and their relationship (constitutive relation) are found in textbooks [1–3].

3.4.3 Torque *T* and Rate of Twist

The approach is to equate the applied torque T to the torque generated in the section by the shear stress distribution. In a small dA as shown in Figure 3.29,

$$dT = \rho \cdot (\tau dA) \tag{3.64}$$

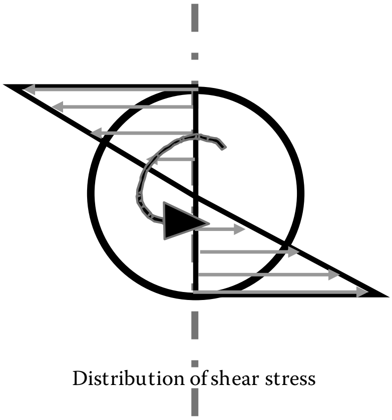


FIGURE 3.28
Stress distribution across a section.

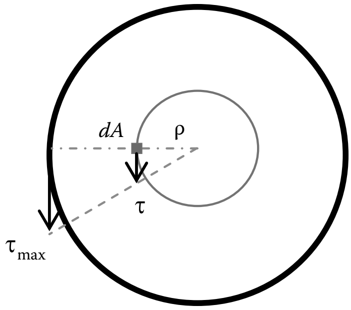


FIGURE 3.29
Small element.

Integrating over the area of the circular beam gives

$$T = \int_A dT = \int_A \rho(\tau\, dA)$$

Because the rate of twist $d\varphi/dx$ is constant through the section, it is not a function of radius ρ. Assuming a homogeneous material, G is also constant, so

$$\begin{aligned} T &= \int_A \rho G\gamma(dA) = G\int_A \rho\cdot\rho\frac{d\varphi}{dx}(dA) \\ &= G\frac{d\varphi}{dx}\int_A \rho^2(dA) = G\frac{d\varphi}{dx}J \qquad (3.65) \\ J &= \int_A \rho^2(dA) \end{aligned}$$

The integral term J is the geometric rigidity of the cross-section and is termed the polar (second) moment of inertia, J.

3.4.4 Shear Stress from Torsion

When we equate Equation 3.65 with the shear stress term, Equation 3.63 gives that [3]

$$\frac{T}{J} = G\frac{d\varphi}{dx} = \frac{\tau}{\rho} \tag{3.66}$$

or

$$\tau = \frac{T\rho}{J} \tag{3.67}$$

where

T the internal torque at the analyzed cross-section
J the shaft's polar moment of inertia
G shear modulus of elasticity for the material
ρ radial distance from the axis (center).

The *maximum shear stress* can be computed as

$$\tau_{\text{max}} = \frac{Tr_{\text{o}}}{J} \tag{3.68}$$

where r_{o} is the radius of the *outer surface of the circular shaft*.

3.5 Stress Transformation in Two Dimensions

3.5.1 Introduction

Previously, we have combined direct stresses due to direct loading and bending in the same, uniaxial, direction. Now we will consider the problems in which stresses in different directions act together and vary throughout a component; that is we are going to find stresses that act on different planes through a point in a material.

However, we cannot work with a point because it has no size, so we work with very small elements that surround the point. We can use small wedges, small cubes, etc. (This work will be restricted to Cartesian coordinates *xyz*.)

To do so, we require rigorous notation for our general system, first considering the most general case in three dimensions and "reducing" this to two dimensions.

3.5.2 General State of Stress in Three Dimensions

In general, stresses may act in a full three-dimensional sense. If we adopt *xyz* coordinates, the resultant forces crossing the surface of an *xyz* cubic element can be resolved

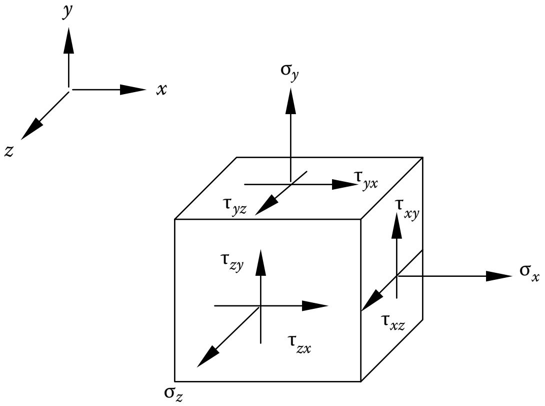


FIGURE 3.30
General state of stress.

and divided by face area to give a set of three-dimensional stresses. These are shown in Figure 3.30 and are often presented in an array:

$$\begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix}$$

However, note that using the concept of complementary shear stress, we get

$$\tau_{xy} = \tau_{yx} \quad \tau_{yz} = \tau_{zy} \quad \tau_{zx} = \tau_{xz}$$

So we find that *six* independent stress values are enough to define a full three-dimensional stress condition, namely σ_x, σ_y, σ_z, τ_{xy}, τ_{yz}, τ_{zx}.

3.5.3 General State of Stress in Two Dimensions

Many practical problems are only two-dimensional or quasi-two-dimensional. In these cases, only three values are needed to define the state of stress. By "two-dimensional," we mean that there are no stresses in one of the three directions (say the z direction). The cubic element of Figure 3.30 can now be represented as a square of unit thickness in the z direction and carrying only x and y stresses as shown in Figure 3.31.

The array reduces to four terms of which three are independent.

$$\begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{yx} & \sigma_y \end{bmatrix}$$

Note, $\tau_{xy} = \tau_{yx}$. Three independent stress values are enough to fully describe a two-dimensional state of stress.

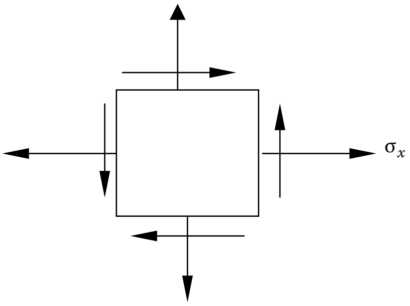


FIGURE 3.31
A 2-D stress element.

3.5.4 Analysis of Plane Stress in Two Dimensions

Assume that σ_x, σ_y, and τ_{xy} are known at a point and use a wedge-shaped element thickness *dz* surrounding the point (Figure 3.32). Use equilibrium of *forces* normal and perpendicular to the face AB to obtain expressions for the two stresses σ_θ, τ_θ on AB.

Forces normal to AB:

$$\sigma_\theta ABdz = \sigma_x BCdz\cos\theta + \sigma_y ACdz\sin\theta + \tau_{xy} BCdz\sin\theta + \tau_{yx} ACdz\cos\theta$$

Using $AC = AB \sin\theta$ and $BC = AB\cos\theta$ and $\tau_{yx} = \tau_{xy}$, we get

$$\sigma_\theta = \sigma_x \cos^2\theta + \sigma_y \sin^2\theta + 2\tau_{xy}\sin\theta\cos\theta$$

Now using $\cos^2\theta = 1/2(1 + \cos 2\theta)$, $\sin^2\theta = 1/2(1 - \cos 2\theta)$ and $2\sin\theta\cos\theta = \sin 2\theta$, the normal stress on *AB* becomes

$$\sigma_\theta = \frac{(\sigma_x + \sigma_y)}{2} + \frac{(\sigma_x - \sigma_y)}{2}\cos 2\theta + \tau_{xy}\sin 2\theta \tag{3.69}$$

Forces parallel to *AB*:

$$\tau_\theta ABdz + \sigma_x BC\,dz\sin\theta - \sigma_y AC\,dz\cos\theta - \tau_{xy} BCdz\cos\theta + \tau_{yx} ACdz\sin\theta = 0$$

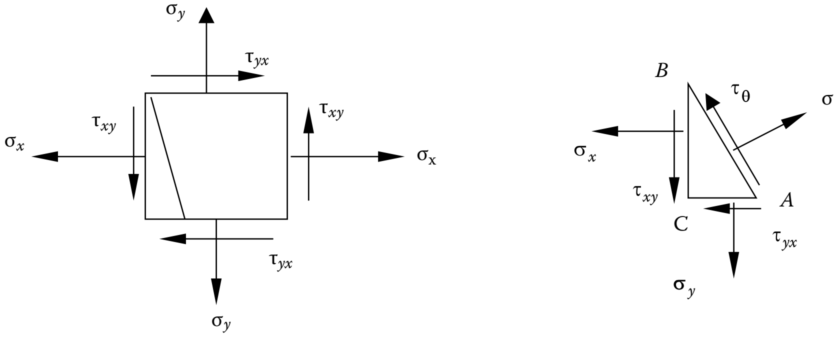

σ_y
τ_{yx}
τ_{xy}
τ_{xy}
σ_x
σ_x
τ_{yx}
σ_y
B
τ_θ
σ_θ
σ_x
τ_{xy}
A
C
τ_{yx}
σ_y

FIGURE 3.32
A 2-D stress element.

That is, $\tau_\theta = -(\sigma_x - \sigma_y)\sin\theta\cos\theta + \tau_{xy}(\cos^2\theta - \sin^2\theta)$ so

$$\tau_\theta = -\frac{(\sigma_x - \sigma_y)}{2}\sin 2\theta + \tau_{xy}\cos 2\theta \tag{3.70}$$

These are the stress transformation equations, starting with XY to get θ.

We can also search through these equations for maximum and minimum values, for example, differentiate Equation 3.69 with respect to θ to find maximum and minimum values of σ_θ:

$$\frac{d\sigma}{d\theta} = \frac{1}{2}(\sigma_x - \sigma_y)(-2\sin 2\theta) + \tau_{xy} 2\cos 2\theta = 0$$

for maximum and minimum, that is,

$$\tan 2\theta = \frac{(2\tau_{xy})}{(\sigma_x - \sigma_y)} \tag{3.71}$$

This has two solutions, θ_1 and θ_2 90° apart; one gives the maximum, and the other gives the minimum. These maximum and minimum values are called *principal stresses* and the planes on which they act are called the *principal plane.*

Similarly, we can differentiate Equation 3.71 to search for maximum and minimum τ_θ

We get

$$\tan 2\theta = -\frac{(\sigma_x - \sigma_y)}{(2\tau_{xy})} \tag{3.72}$$

There are two solutions to this equation, θ_3 and θ_4, which are also 90° apart and lie halfway (±45°) between the principal planes. These are called the planes of maximum shear. Now let us see if it ever has a zero value, that is, put τ_θ Equation 3.70 = 0.

This gives

$$\tan 2\theta = \frac{(2\tau_{xy})}{(\sigma_x - \sigma_y)}.$$

Notice that this is the same result as Equation 3.71; therefore, there is no shear on the principal planes. So we have discovered how two-dimensional stress varies according to the plane that we choose as the reference.

There will always be principal planes and maximum shear planes. The principal stresses will be the largest and smallest values of the normal stress σ on any plane through the point in the component and will therefore be larger and smaller than σ_x and σ_y (unless σ_x and σ_y themselves are the principal stresses).

Also there will exist $\tau_{\max} > \tau_{xy}$ (unless $\tau_{xy} = \tau_{\max}$). So, in any analysis or design, it will be necessary to find the principal stresses and the max shear stress because they will be the significant values of stress as well as the ones to compare with the design stress allowables.

Now we substitute Equation 3.71 into Equation 3.69 and then Equation 3.72 into Equation 3.70 to get expressions for the values of the principal stresses and max shear stress:

$$\sigma_1 = \frac{(\sigma_x + \sigma_y)}{2} + \frac{1}{2}\sqrt{(\sigma_x - \sigma_y)^2 + 4\tau_{xy}{}^2} \tag{3.73}$$

$$\sigma_2 = \frac{(\sigma_x + \sigma_y)}{2} - \frac{1}{2}\sqrt{(\sigma_x - \sigma_y)^2 + 4\tau_{xy}{}^2} \tag{3.74}$$

$$\tau_{\max} = \frac{1}{2}\sqrt{(\sigma_x - \sigma_y)^2 + 4\tau_{xy}{}^2} \tag{3.75}$$

Can you prove that $\tau_{\max} = \frac{1}{2}(\sigma_1 - \sigma_2)$?

3.5.5 Calculation of Strains from Stresses

3.5.5.1 Hooke's Law

E = Young's modulus

ν = Poisson's ratio

Three dimensional form, that is, three principal stresses σ_1, σ_2, and σ_3

$$\begin{aligned} \varepsilon_1 &= \frac{\sigma_1}{E} - \nu\frac{\sigma_2}{E} - \nu\frac{\sigma_3}{E} \\ \varepsilon_2 &= \frac{\sigma_2}{E} - \nu\frac{\sigma_3}{E} - \nu\frac{\sigma_1}{E} \\ \varepsilon_3 &= \frac{\sigma_3}{E} - \nu\frac{\sigma_1}{E} - \nu\frac{\sigma_2}{E} \end{aligned} \tag{3.76}$$

Two dimensional form, that is, two principal stresses σ_1 and σ_2.

$$\begin{aligned} \varepsilon_1 &= \frac{\sigma_1}{E} - \nu\frac{\sigma_2}{E} \\ \varepsilon_2 &= \frac{\sigma_2}{E} - \nu\frac{\sigma_1}{E} \\ \varepsilon_3 &= -\nu\frac{\sigma_1}{E} - \nu\frac{\sigma_2}{E} \end{aligned} \tag{3.77}$$

One dimensional form, that is, one principal stress σ_1.

$$\varepsilon_1 = \frac{\sigma_1}{E}$$

$$\varepsilon_2 = -\upsilon\frac{\sigma_1}{E} \tag{3.78}$$

$$\varepsilon_3 = \upsilon\frac{\sigma_1}{E}$$

3.6 Strain Analysis and the Strain Gauge Rosettes

3.6.1 Introduction

We have already seen that the description of elastic stresses must contain a defined reference plane and that the magnitude and direction of the stresses change as we change this reference plane.

Because elastic stresses and strains are related by Hooke's Law, which is linear, it is expected that an analysis of strains may be developed that is very similar to the two-dimensional stress analysis [3]. This is indeed so.

The starting point would be a knowledge of the direct strains and shear strain referred to predefined x and y directions. These would be ε_x, ε_y, and γ_{xy}.

The analysis is based on the geometry of the strained body, and a first result is to discover the strains referred to a pair of *axes* θ and θ + 90°, which are located at θ from the x and y axes.

The expressions for these strains are

$$\varepsilon_\theta = \varepsilon_x \cos^2\theta + \varepsilon_y \sin^2\theta + \gamma_{xy} \sin\theta\cos\theta \tag{3.79}$$

$$\varepsilon_{\theta+90} = \varepsilon_x \sin^2\theta + \varepsilon_y \cos^2\theta - \gamma_{xy}\sin\theta\cos\theta \tag{3.80}$$

$$\gamma_{\theta,\theta+90} = \gamma_{xy}\left(\cos^2\theta - \sin^2\theta\right) - 2(\varepsilon_x - \varepsilon_y)\sin\theta\cos\theta \tag{3.81}$$

These equations have exactly the same form as those describing the variation of stress as the reference *plane* is rotated. So it would be possible to search them, say, for maximum and minimum values of strain, etc. However, it is not really necessary to go through all this because we can anticipate the results from our knowledge of stress analysis.

We will find that there are principal values of direct strain ε_1 and ε_2 (maximum and minimum, respectively), which occur in directions at right angles to each other. Also there is a zero shear strain between the principal directions 1 and 2. Also that the shear strain has a maximum value between two directions which are at 45° to the principal directions.

All this is looking familiar, but we will come to the stage soon in which we have to put real values of strain into the expressions and do some calculations. And that is where a big snag comes in. While direct strains (ε) are easy to measure accurately, for example, by the use of strain gauges, shear strain (γ) is extremely difficult to obtain by measurement.

The solution to this problem is to reconstruct the equations of strain by eliminating all shear strains. We do this by "swapping" shear strain for an extra reading of direct strain.

That is,

Data set	$\varepsilon_x \ \varepsilon_y \ \gamma_{xy}$	Easy to use	Difficult to obtain
Data set	$\varepsilon_A \ \varepsilon_B \ \varepsilon_C \ \theta$	Usable	Easy to obtain

So we make three measurements of direct strain in three known directions and use them in modified equations or a modified Mohr's circle. We could choose any directions at all, but standard practice in strain measurement has been settled on two choices for practical use.

3.6.2 Strain Gauge Rosettes

These usually consist of three linear strain gauges mounted close to each other on the same base pad so that they measure strains in three known directions [3].

We will deal with them in turn because they each have their own form of the equations. The useful equations are those for the two principal strains ε_1 and ε_2 and for the maximum shear strain $\gamma_{\max}$.

Rectangular rosettes

A is at 0°

$$\varepsilon_1 = \frac{(\varepsilon_A + \varepsilon_C)}{2} + \frac{1}{2}\sqrt{(\varepsilon_A - \varepsilon_C)^2 + (\varepsilon_A + \varepsilon_C - 2\varepsilon_B)^2} \tag{3.82}$$

B is at 45°

$$\varepsilon_2 = \frac{(\varepsilon_A + \varepsilon_C)}{2} - \frac{1}{2}\sqrt{(\varepsilon_A - \varepsilon_C)^2 + (\varepsilon_A + \varepsilon_C - 2\varepsilon_B)^2} \tag{3.83}$$

C is at 90°

$$\frac{1}{2}\gamma_{\max} = \frac{1}{2}\sqrt{(\varepsilon_A - \varepsilon_C)^2 + (\varepsilon_A + \varepsilon_C - 2\varepsilon_B)^2} \tag{3.84}$$

Delta rosettes

A is at 0°

$$\varepsilon_1 = \frac{(\varepsilon_A + \varepsilon_B + \varepsilon_C)}{3} + \frac{1}{3}\sqrt{2\{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_B - \varepsilon_C)^2 + (\varepsilon_C - \varepsilon_A)^2\}} \tag{3.85}$$

B is at 60°

$$\varepsilon_2 = \frac{(\varepsilon_A + \varepsilon_B + \varepsilon_C)}{3} - \frac{1}{3}\sqrt{2\{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_B - \varepsilon_C)^2 + (\varepsilon_C - \varepsilon_A)^2\}} \tag{3.86}$$

C is at 120°

$$\frac{1}{2}\gamma_{\max} = \frac{1}{3}\sqrt{2\{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_B - \varepsilon_C)^2 + (\varepsilon_C - \varepsilon_A)^2\}} \tag{3.87}$$

3.6.3 Conversion from Principal Strains to Principal Stresses

The principal strains obtained from the graphical calculation methods are used to calculate the principal stresses using Hooke's Law with $\sigma_3 = 0$.

$$\begin{aligned} \varepsilon_1 &= \frac{\sigma_1}{E} - \nu\frac{\sigma_2}{E} \\ \varepsilon_2 &= \frac{\sigma_2}{E} - \nu\frac{\sigma_1}{E} \\ \varepsilon_3 &= -\nu\frac{\sigma_1}{E} - \nu\frac{\sigma_2}{E}\,. \end{aligned} \tag{3.88}$$

These give

$$\begin{aligned} \sigma_1 &= E\frac{(\varepsilon_1 + \nu\varepsilon_2)}{(1-\nu^2)} \\ \sigma_2 &= E\frac{(\varepsilon_2 + \nu\varepsilon_1)}{(1-\nu^2)} \end{aligned} \tag{3.89}$$

As before

$$\tau_{\max 12} = \frac{(\sigma_1 - \sigma_2)}{2} \tag{3.90}$$

3.7 Mechanical Properties of Materials

3.7.1 Introduction

Before the theoretical stress analysis of a projected component can be completed, we must know some information about the properties of materials, such as strength, Young's modulus, from which the component will be made. A manufacturer engaged in producing a component from the raw material into finished geometry or dimension has to know whether the mechanical properties are up to specification. Thus, a knowledge of mechanical tests and properties of materials is essential to an engineer.

3.7.2 Tension and Compression Tests

The tension and compression tests are the most widely used methods in order to obtain the stress–strain relationship of many engineering materials, such as metals, polymers, ceramics, and composites. As defined by Hooke in 1678, straining a bar in tension and measuring the resulting force is termed a tensile test.

As shown in Figure 3.33, a specimen of material is made into a "standard" shape and size to perform either a tension or compression test. The sample can be tested in a standard tensile test machine as shown in Figure 3.34.

3.7.3 Stress–Strain Behavior of Ductile Materials

From the data of a test, it is possible to determine various values of the stress and corresponding strain in the specimen and plot the results. The resulting curve is called the stress–strain diagram, which describes the relationship between stresses and strains. Figure 3.35 shows a typical schematic view of a stress–strain diagram in which there are three key values in the stress–strain curve: elastic limit stress σ_{el}, proportional limit stress σ_{pl}, and ultimate limit σ_{ult}.

It can be seen that stress is proportional to strain with an increase of the strain or load within σ_{el}, but stress on the curve will be back to zero when the strain or load is back to zero. There is no permanent strain remaining. This stress is termed the elastic limit σ_{el}. The slope in the elastic curve is called Young's modulus E, which is of importance in engineering. A relationship between stress and strain in the elastic region is defined by Hooke's law as follows:

$$\sigma = E\varepsilon \tag{3.91}$$

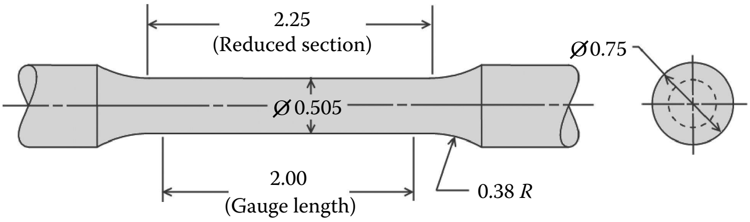


FIGURE 3.33
A standard tensile test specimen (dimension inches).

FIGURE 3.34
A typical tensile machine.

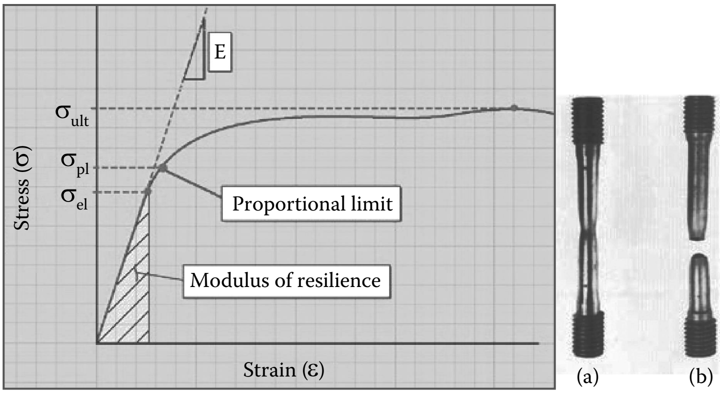

E
σ_{ult}
σ_{pl}
σ_{el}
Stress (σ)
Proportional limit
Modulus of resilience
Strain (ε)
(a)
(b)

FIGURE 3.35
A typical schematic view of stress–strain diagram (a) Necking, (b) Fracture.

where *E* is Young's modulus, σ is stress, and ε is strain. Hooke's law is the foundation of material property. A detailed treatment of this subject is found in [1–3].

Beyond the elastic limit σ_{el}, is a plastic region in which a permanent deformation remains after a load is removed. The maximum stress in Figure 3.35 is called ultimate stress σ_{ult} because the material will have either further deformation/strain or fracture with an increase of loading. Between the elastic and ultimate stresses, the stress is termed the proportional limit σ_{pl}, which indicates the stress with a tolerant and limited permanent deformation after a load is removed. σ_{pl} is also termed yield stress.

3.7.4 Poisson's Ratio

As shown in Figure 3.36, when a deformable component is subjected to an axial tensile force, not only does it elongate, but it also contracts laterally. Similarly, a compressive force acting on the component results in contraction in the loading direction, but its sides expand laterally. In Figure 3.36, when the load F is exerted on the bar, it changes the length of the bar from l_o to l_i and its radius from d_o to d_i. The strains in the longitudinal or axial direction and in the lateral or radius direction are, respectively, the longitudinal strain

$$\varepsilon_{\text{long}} = \frac{l_i - l_o}{l_o}$$

and the lateral strain

$$\varepsilon_{\text{lateral}} = \frac{d_i - d_o}{d_o}$$

The ratio of these strains in both directions can be defined as Poisson's ratio

$$\nu = \frac{\varepsilon_{\text{lateral}}}{\varepsilon_{\text{long}}} \tag{3.92}$$

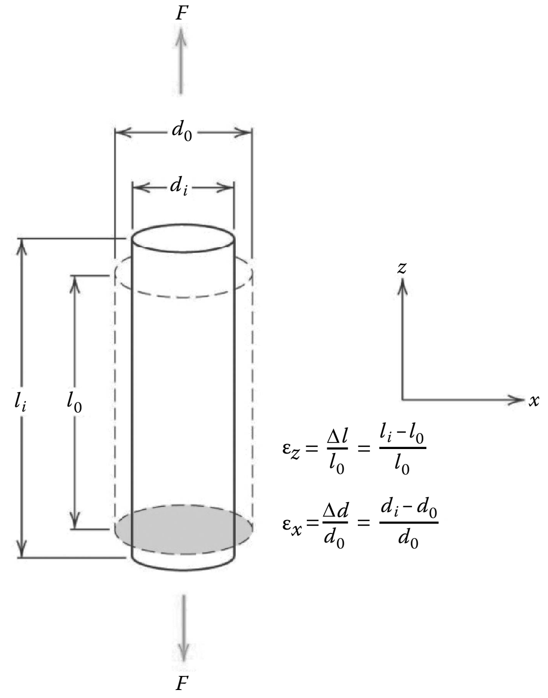

F
d_0
d_i
z
x
l_i
l_0
$\varepsilon_z = \frac{\Delta l}{l_0} = \frac{l_i - l_0}{l_0}$
$\varepsilon_x = \frac{\Delta d}{d_0} = \frac{d_i - d_0}{d_0}$
F

FIGURE 3.36
Two types of elasticity.

Poisson's ratio ν is dimensionless and constant for a given material. For most nonporous solid materials, the values of Poisson's ratio ν can be found from a handbook of manufacturing.

3.8 Conclusions

This chapter presented the fundamental theory of mechanics of materials, which can be employed for an application in the field of mechatronics. The chapter started with three basic cases: axial loading, bending, and torsion. Then stress and strain were included and followed by analysis of displacements. Both stress and strain transformation were covered with strain gauge rosettes for a measurement purpose in mechatronics. A brief section of mechanical tests and properties of materials that are essential to an engineer have also been introduced.

References

1. Hibbeler, R. C., *Mechanics of Materials*, Pearson Prentice Hall, New Jersey, 2005.
2. Benham, P. O., R. J. Crawford, and C. G. Armstrong, *Mechanics of Engineering Materials*, Longman, Harlow, England, 1996.
3. De Silva, C. W., *Mechanics of Materials*, Taylor & Francis, CRC Press, Boca Raton, FL, 2015.

4

Control of Mechatronic Systems

Kok Kiong Tan and Andi Sudjana Putra

CONTENTS

4.1 What Is a Mechatronic System? 86
4.2 Overview of Control Systems 88
  4.2.1 System Model 88
  4.2.2 System Modeling Applied to Components of Mechatronic Systems 90
  4.2.3 Performance Assessment of a Control System 92
4.3 Control Techniques 93
  4.3.1 Feedback Proportional–Integral–Derivative (PID) Control 93
  4.3.2 Feedforward Control 101
  4.3.3 Servo Control Structures 101
  4.3.4 Programmable Logic Controllers 102
4.4 Implementation of a Computer Control 103
4.5 Challenges in Control of Mechatronic Systems 106
  4.5.1 Friction 106
  4.5.2 Force Ripples 107
  4.5.3 Hysteresis and Backlash 108
  4.5.4 Saturation 109
  4.5.5 Dead Zone 109
  4.5.6 Reference Signal Changes 109
  4.5.7 Low-Frequency Drift 110
  4.5.8 High-Frequency Noise 110
  4.5.9 Incorporating and Addressing Nonlinear Dynamics 111
4.6 Application Examples 114
  4.6.1 Flight Simulators 114
  4.6.2 Piezoelectric Control System for Biomedical Application 116
4.7 Conclusions 122
Bibliography 123

SUMMARY This chapter addresses control issues in mechatronic systems, including various examples and implementations of the control systems for illustration as well as standards and guides to common control software and algorithms. It will begin with a discussion on what mechatronic systems are and what has made them gain widespread use in less than 40 years with some illustrative examples. A brief overview of control systems is then presented with some techniques of control that are necessary for subsequent discussions on the control of mechatronic systems. The focus of the discussions is on implementation issues, which include how to implement a practical technique and the challenges in implementations. Case studies—from real implementations—given at the end of this chapter provide a glimpse at the control of mechatronic systems in practice.

4.1 What Is a Mechatronic System?

The term *mechatronic* was coined by Tetsura Mori from the Yasakawa Electric Corp. in 1969 as a contraction of "mecha" from mechanism and "tronic(s)" from electronic(s). There are many definitions of mechatronic, one of which is the "synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes" (by Harashima, Tomizuka, and Fukuda, 1996). For a system to qualify as a mechatronic system, it must have the following constituent components:

1. Mechanical elements, such as actuators, machines, mechanics, etc.
2. Electronic elements, such as sensors, circuits, computation hardware, etc.
3. Information/programming elements, such as control software, artificial intelligence, etc.

An example of a mechatronic system is the automated hospital bed. As patients might have various difficulties or complications with their position and movement, hospital beds are designed to allow actuator-assisted movements for various patients' positions, such as reclining, knee-bending, elevation, etc. The actuators are typically AC motors* to match with the electrical power source available in hospitals. The AC motors are controlled by humans, either the patients or caregivers, using a handheld controller that has a few buttons, each of which corresponds to a particular movement or direction.

Between the handheld controller and the AC motors, however, are myriad circuits and a control system to regulate the movement of the bed. The control system of the bed translates the occasionally trembling, intermittent, or unregulated pushes of a button that patients might give into smooth, regulated movements that patients need. Such regulated movements from unregulated inputs are achieved through computation, filtering, sensing, and other functions that form the backbone of a mechatronic system control, as presented in Figure 4.1.

The birth of mechatronics has revolutionized many products and services and has brought about a change to the perspective of engineering design. It changes the approach engineers take in designing solutions to a set of problems from designing the mechanism and *then* the electronics to house countrollers to designing the mechanism *concurrently* with designing its electronics. In the historical perspective, it is a new approach of engineering design as illustrated in Figure 4.2. It was the invention of electromagnetic devices that first brought about the possibility of marrying mechanical and electrical systems although it was not integrative at the initial development. Electric motors, due to their ease of maintenance and size, were used to replace prime movers, such as steam engines, and in that case, simple replacement was all that was needed. The invention of relays and solenoids (and their derivatives, such as solenoid valves) brought a new dimension to the use of electric devices for control with their ability to convert electric signals into mechanical actions. Their application, in turn, was elevated to another level with the computing power of computers to realize complex control algorithms, making way for the synergistic design that gave birth to mechatronic systems.

A special type of mechatronic system in which the emphasis lies on the control component is a servo system, defined as a system that is able to control some variables of interest

---

* AC, or alternating current, motors are types of electric motors that run on alternating current supply.

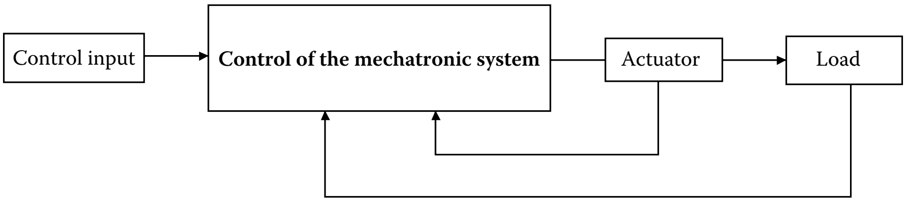

Control input
Control of the mechatronic system
Actuator
Load

FIGURE 4.1
Control structure of a mechatronic system. The output of the control system is monitored by sensing the action of the actuators or, even more downstream, of the load.

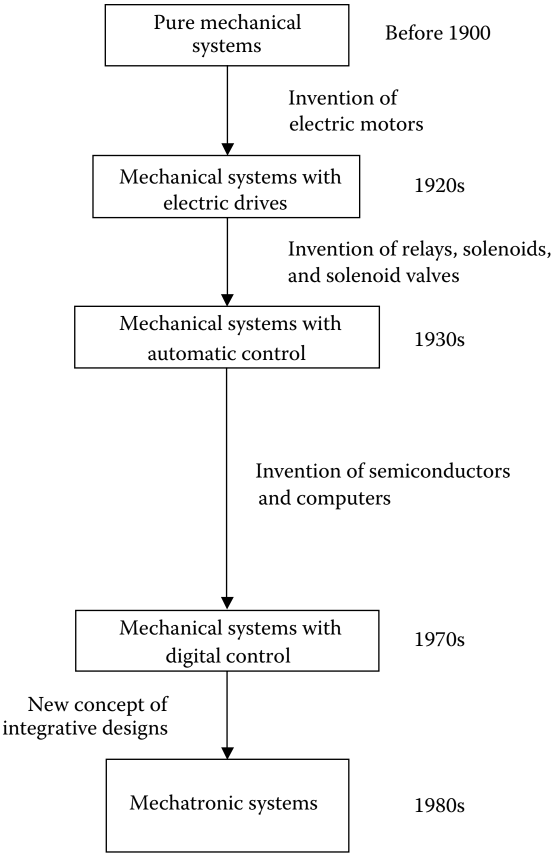

Pure mechanical systems
Before 1900
Invention of electric motors
Mechanical systems with electric drives
1920s
Invention of relays, solenoids, and solenoid valves
Mechanical systems with automatic control
1930s
Invention of semiconductors and computers
Mechanical systems with digital control
1970s
New concept of integrative designs
Mechatronic systems
1980s

FIGURE 4.2
Advent in technology leading to the advancement of mechatronics.

to track user-specified objectives closely and whose control is termed servo control. Today, servo control has become an integral part of almost every automation system or process, including those found in manufacturing, chemical, petrochemical, transportation, military, and biomedical areas.

A mechatronic system therefore spans the spectrum of mechanical, electrical, and programming. These three elements, working synergistically, are what make a system "mechatronic." As illustrated in Figure 4.3, mechatronics sits at the center of these three elements.

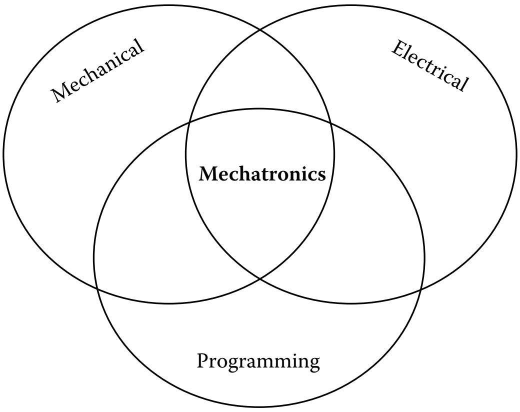


FIGURE 4.3
Mechatronics at the intersection of mechanical, electrical, and programming.

4.2 Overview of Control Systems

The controller orchestrates the different components of a mechatronic system to yield desired performance. The design and analysis of the controller are based on constituent fields of system theory, a condensed part of which is discussed in this section.

4.2.1 System Model

The analysis of a system can be performed using its model, which can be obtained via a physical or system identification approach. A system identification approach of a hard disk drive (HDD), as an example, will give a response plot,* such as the one in Figure 4.4, which can be used to identify the system model.

The same model can be obtained using physical modeling as depicted in Figure 4.5. With this approach, the model of the system is obtained through a mathematical derivation.

A system can be described in various forms, such as the following:

1. Transfer function
2. Zero-pole-gain
3. State space

Using a **transfer function**, a system is described as a Laplace transform (i.e., in the frequency domain) of the output to the input as follows:

$$G(s)=\frac{b_0s^m+b_1s^{m-1}+b_2s^{m-2}+\ldots+b_m}{a_0s^n+a_1s^{n-1}+a_2s^{n-2}+\ldots+a_n} \tag{4.1}$$

---

* The response plot provides information about the system of interest's output with respect to known input. These response plots can be in the time or frequency domain, either of which will give the identical model.

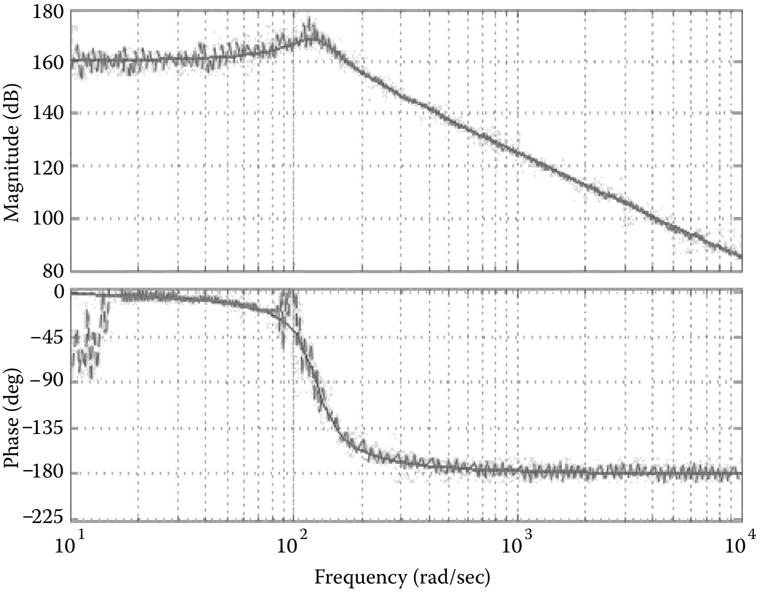


FIGURE 4.4
Frequency response of a HDD from a system identification experiment. (Dashed line: frequency response from experiment, solid line: frequency response of the model.)

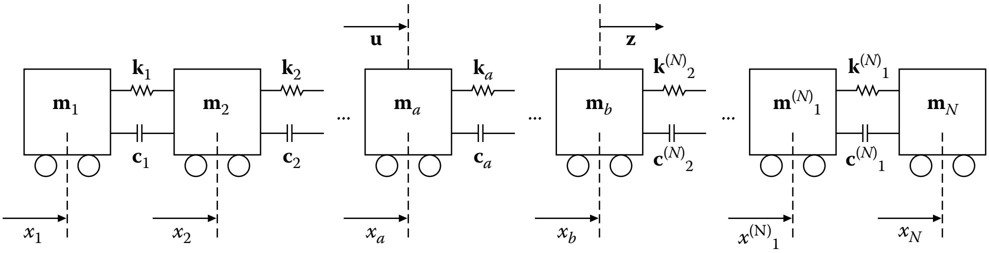

u
z
$\mathbf{k}_1$
$\mathbf{k}_2$
$\mathbf{k}_a$
$\mathbf{k}^{(N)}{}_2$
$\mathbf{k}^{(N)}{}_1$
$\mathbf{m}_1$
$\mathbf{m}_2$
$\mathbf{m}_a$
$\mathbf{m}_b$
$\mathbf{m}^{(N)}{}_1$
$\mathbf{m}_N$
$\mathbf{c}_1$
$\mathbf{c}_2$
$\mathbf{c}_a$
$\mathbf{c}^{(N)}{}_2$
$\mathbf{c}^{(N)}{}_1$
x_1
x_2
x_a
x_b
$x^{(N)}{}_1$
x_N

FIGURE 4.5
Physical model of an HDD, in which **u** represents the force and **z** represents the position; with **m**, **k**, and **c** representing the mass, elasticity constant, and damping constant, respectively.

where $b_0, b_1, b_2, \ldots, b_m$ and $a_0, a_1, a_2, \ldots, a_m$ are the coefficients of the polynomials of the Laplace numerator and denominator, respectively. A system identification approach to identify a system model can provide the model in a transfer function form. Referring to the HDD physical model as presented in Figure 4.5, the transfer function of the HDD is as follows:

$$P(s) = \frac{2.861 \times 10^{21}}{s^4 + 867.07s^3 + 1.6681 \times 10^9 s^2 + 8.3863 \times 10^{10} s + 2.6338 \times 10^{13}} \tag{4.2}$$

A transfer function can also be posed in a **zero-pole-gain** form as follows:

$$G(s) = K \frac{\prod_{i=1}^{m} (s - z_i)}{\prod_{i=1}^{n} (s - p_I)} \tag{4.3}$$

where K is the gain of the system, z_i is the ith zero, and p_i is the ith pole. The system model in Equation 4.2 can therefore be rewritten in a zero-pole-gain form as follows:

$$G(s) = 2.861 \times 10^{21} \frac{1}{(s+408-40840i)(s+408+40840i)(s+25-123i)(s+25+123i)} \tag{4.4}$$

A system can also be described using a state variable description, called **state space**, which is a collation of differential equations as follows:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \tag{4.5}$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \tag{4.6}$$

where **x** is a column vector ($n \times 1$) representing the states of an nth-order system, $\dot{\mathbf{x}}$ is the time-derivative of **x**, **u** is a column vector ($m \times 1$) representing the inputs to the system, **y** is a column vector ($p \times 1$) representing the output of the system, **A** is the system matrix ($n \times n$), **B** is the input matrix ($n \times m$), **C** is the output matrix ($p \times n$), and **D** is the feed-through matrix ($p \times m$). Using the same HDD model as in Equations 4.2 and 4.4, the state space model of the HDD is as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -0.0000 & -0.0002 & -0.0084 & -2.6338 \\ 0.0000 & 0 & 0 & 0 \\ 0 & 0.0000 & 0 & 0 \\ 0 & 0 & 0.0000 & 0 \end{bmatrix} \times 10^{13} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{4.7}$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 2.8610 \end{bmatrix} \times 10^{21} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{4.8}$$

4.2.2 System Modeling Applied to Components of Mechatronic Systems

As discussed in Section 4.1, a mechatronic system spans the mechanical, electrical, and programming spectra, in a configuration of sensors, controllers, and actuators as presented in Figure 4.3. Each sensor or actuator is also a system in itself and can therefore be modeled. A thermocouple, for example (schematic in Figure 4.6), can be modeled as follows:

$$V(T) = a_1 T + a_2 T^2 + \ldots + a_n T^n \tag{4.9}$$

where V is the voltage signal generated by the thermocouple, a_n the thermocouple coefficients, and T the temperature.

Similarly, actuators also possess their own dynamics, and they can be seen as systems. The dynamics of the actuator can be viewed as comprised of two components: a dominantly linear model and an uncertain and nonlinear remnant, which we will discuss in

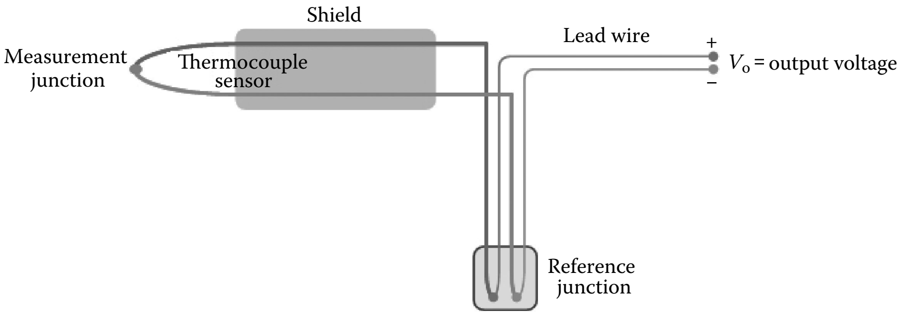


FIGURE 4.6
Thermocouple schematic.

Section 4.5. In this section, we first discuss the linear dynamics of actuators, which mainly depends on the physical behavior of the actuation mechanisms.

For illustration purposes, consider an electric motor in a mechatronic system. In the dominant linear model, the mechanical and electrical dynamics of the motor can be expressed as follows:

$$I\ddot{\theta} + D\dot{\theta} + \tau_{\text{load}} = \tau_{\text{M}} \tag{4.10}$$

$$K_{\text{e}}\dot{\theta} + L_{\text{a}}\frac{\text{d}i_{\text{a}}}{\text{d}t} + R_{\text{a}}i_{\text{a}} = u \tag{4.11}$$

$$\tau_{\text{M}} = K_{\text{t}}i_{\text{a}} \tag{4.12}$$

where θ, I, D, τ_{M}, and τ_{load} denote the mechanical parameters of position, inertia, viscosity constant, generated torque, and load torque, respectively, and u, i_{a}, R_{a}, and L_{a} denote the electrical parameters: input DC voltage,* armature current, armature resistance, and armature inductance, respectively, and K_{t} denotes an electrical–mechanical energy conversion constant as presented in Figure 4.7. The variable J in Figure 4.7 represents the inertia of the motor.

Because the electrical time constant is typically much smaller than the mechanical time constant, the delay due to electrical transient response may be ignored, giving the following simplified model:

$$\ddot{\theta} = -\frac{K_1}{I}\dot{\theta} + \frac{K_2}{I}u - \frac{1}{I}\tau_{\text{load}} \tag{4.13}$$

where

$$K_1 = \frac{K_{\text{e}}K_{\text{t}} + R_{\text{a}}D}{R_{\text{a}}} \tag{4.14}$$

$$K_2 = \frac{K_{\text{t}}}{R_{\text{a}}} \tag{4.15}$$

---

[*] DC, or direct current, refers to the type of current that does not change polarity periodically.

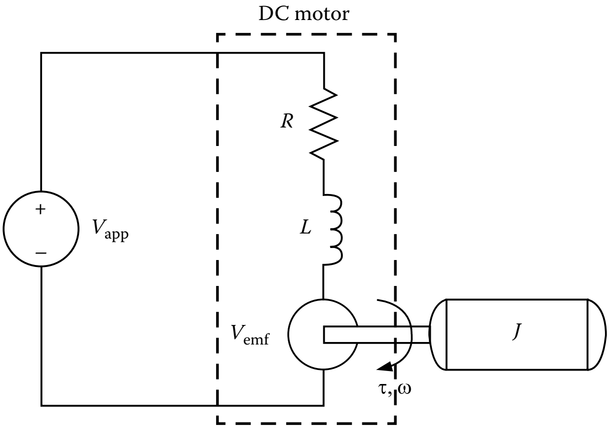


FIGURE 4.7
Electric representation of a DC motor system.

In this model, the input u of the motor is V_{app}, and the output is the rotational speed ω. Therefore, the problem of interest of determining V_{app} to achieve certain ω is translated into determining u to achieve certain $\dot{\theta}$.

4.2.3 Performance Assessment of a Control System

The representative behavior of a stable system, upon the introduction of step input, is presented in Figure 4.8 in which K_C is the gain. The characteristics of the system, for example, overshoot (M_p), rise time (t_r), and settling time (t_s), depend on the parameters of the system.

The role of motion control is to force these characteristics to meet the required specifications so that the system can deliver the required output satisfactorily, for example, with a certain overshoot or settling time. Suppose the transfer function of the actuator is $G_{act}(s)$

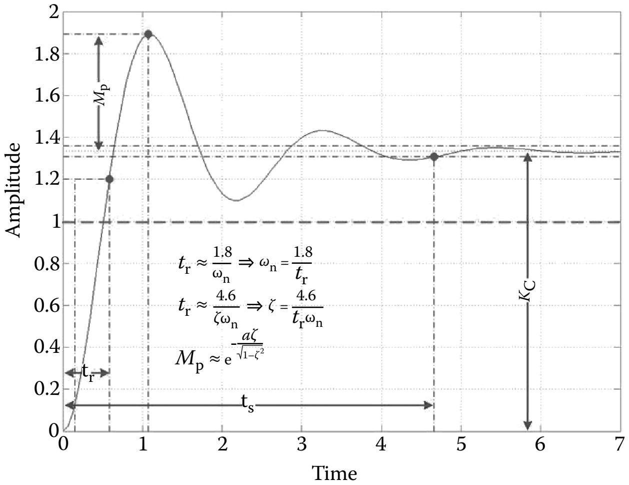


FIGURE 4.8
Step response of a second-order system.

and the combined sensor and control is $G_{\text{ctr}}(s)$, the transfer function of the system can be expressed as

$$\frac{Y(s)}{R(s)} = \frac{G_{\text{ctr}}(s)G_{\text{act}}(s)}{1 + G_{\text{ctr}}(s)G_{\text{act}}(s)} \tag{4.16}$$

showing that the output of the system is affected by $G_{\text{ctr}}(s)$.

4.3 Control Techniques

4.3.1 Feedback Proportional–Integral–Derivative (PID) Control

PID control remains widely used in the industry today even after being around for more than 70 years with applications ranging from precision motion control for assembly and process automation to process condition regulation. The wide usage of PID controllers can be attributed to their reliability, simplicity, and successful applications.

The structure of PID control consists of three constituent components: namely, the proportional, integral, and derivative components. These three components are independent of each other and yet can be combined by simple summation.

The proportional control is one with which the control signal is *proportional* to the error signal such that the control signal is proportionally small when the error signal is small and high when the error signal is high. It can be expressed with the following equation:

$$u = \begin{cases} u_{\min}, & e \le -e_0 \\ u_0 + K_P e, & -e_0 < e < e_0 \\ u_{\max}, & e_0 \le e \end{cases} \tag{4.17}$$

where K_p is the proportional gain, and u_0 is the level of control signal when there is no control error. $u_{\min}$ and $u_{\max}$ are included to account for the realistic limitation of control signals. K_p is the control parameter that indicates the change in the control signal per unit change in the error signal and which may be adjusted by the operator to achieve the desired performance. Figure 4.9 illustrates the effect of proportional control and of proportional gain when an unstable system whose transfer function is $G(s) = \dfrac{2}{s^2 + 2s - 3}$ is controlled using purely proportional control of K_p of 1.0, 3.0, and 5.0, respectively.

As evident from the above equation, there is bound to be a steady state error when a pure proportional control is employed.* It is therefore customary to use proportional control in combination with other control components.

The integral control is one with which the control signal is the accumulation, that is, integration, of the past error signals. The combination of integral and proportional control,

---

* With simple derivation, the error equation of $e = \dfrac{u - u_0}{K_P}$ is obtained. Zero steady state error implies that for the error to be zero, either K_P is to be infinitely large or u to be zero, neither of which is realistic.

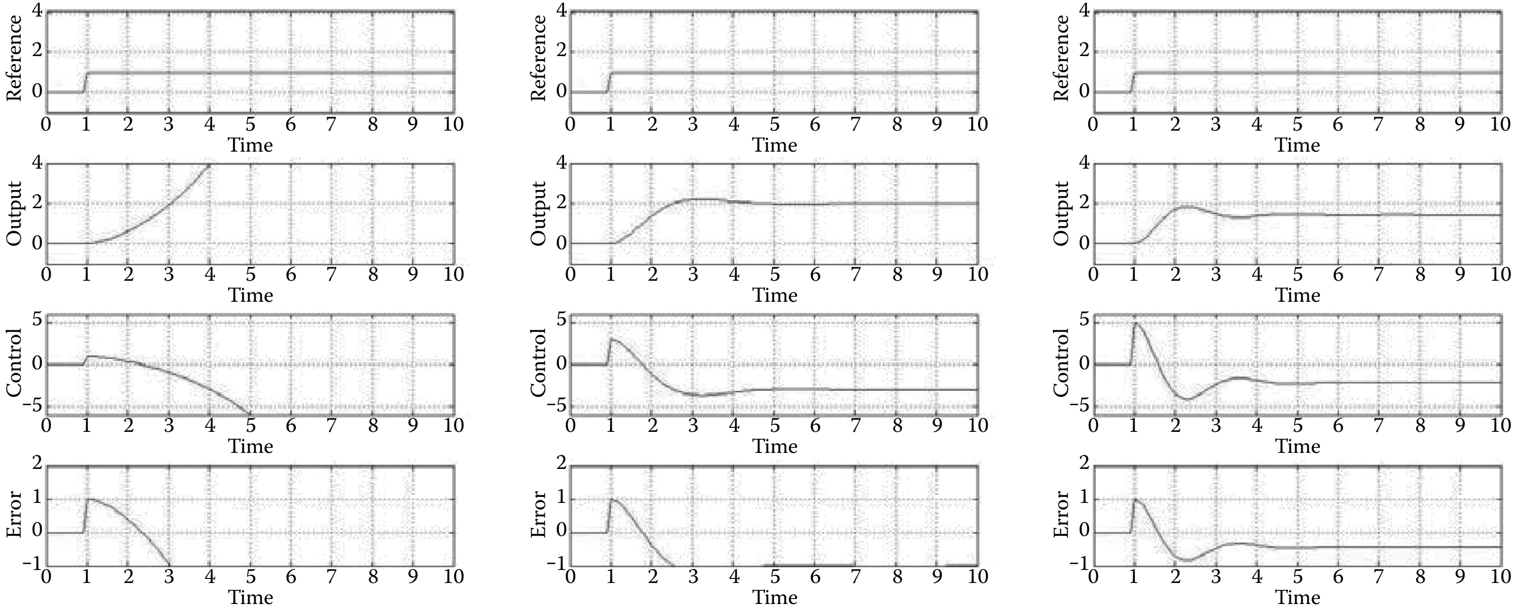


FIGURE 4.9
Performance of a proportional control of K_p of 1.0, 3.0, and 5.0 to $G(s)$.

known as PI control, resolves the steady state error problem of pure proportional control without excessive control gain. The control signal of a PI control is as follows:

$$u = K_{\mathrm{P}}\left(e + \frac{1}{T_{\mathrm{I}}}\int e\,\mathrm{d}t\right) \tag{4.18}$$

where T_{I} is the integral time of the controller. The factor $K_{\mathrm{p}}/T_{\mathrm{I}}$ is customarily known as the integral gain, K_{I}. Figure 4.10 illustrates the effect of PI control and of integral gain with which an unstable system whose transfer function is $G(s) = \dfrac{2}{s^2 + 2s - 3}$ is controlled using PI control K_{p} of 5.0 with K_{I} of 0.0 (i.e., purely proportional control), 1.0, and 3.0, respectively. It demonstrates the steady state error elimination with the introduction of an integral gain.

The derivative control is one with which the control signal is proportional to the current change of error signal, that is, to the time *derivative* of the error. Although it is theoretically possible to have a pure derivative control action, it is not practical because, if the error is large but unchanging, the control output will be erroneously zero. As such, derivative control is usually combined with proportional control to form PD control or even with proportional and integral control to form PID control with the following control law:

$$u = K_{\mathrm{P}}\left(e + \frac{1}{T_{\mathrm{I}}}\int e\,\mathrm{d}t + T_{\mathrm{D}}\frac{\mathrm{d}e}{\mathrm{d}t}\right), \tag{4.19}$$

where T_{D} is the derivative time of the controller. The factor $K_{\mathrm{P}}T_{\mathrm{D}}$ is customarily known as the derivative gain K_{D}. The more practical approach is to replace the error signal in the derivative term with the output signal, considering that $\dfrac{\mathrm{d}e}{\mathrm{d}t} = -\dfrac{\mathrm{d}y}{\mathrm{d}t}$. This is because the reference signal is usually constant with occasional abrupt changes, making the $\dfrac{\mathrm{d}e}{\mathrm{d}t}$ change too much when abrupt changes take place. The PID control law of this approach is as follows:

$$u = K_{\mathrm{P}}\left(e + \frac{1}{T_{\mathrm{I}}}\int e\,\mathrm{d}t - T_{\mathrm{D}}\frac{\mathrm{d}y}{\mathrm{d}t}\right). \tag{4.20}$$

Figure 4.11 illustrates the effect of PID control and of the derivative gain, in which an unstable system whose transfer function is $G(s) = \dfrac{2}{s^2 + 2s - 3}$ is controlled using PID control of K_{P} of 5.0 and K_{I} of 3.0 with K_{D} of 0.0 (i.e., PI control), 1.0, and 2.0, respectively. It demonstrates the improvement of error prediction to reduce oscillation.

As these PID gains affect the performance of a system, tuning these gains, that is, determining their values, are of importance to control a system to achieve the desired performance. In general, the dilemma of tuning the PID gains is the compromise between speed and stability. Fast control is usually accompanied by poor stability and oscillations. A step change in the set point may well result in a severe overshoot. On the other hand, very stable control without overshoot is usually achieved at the expense of a more sluggish response to set point changes and load disturbances. The solution to this compromise may also depend on the type of disturbance present in the application.

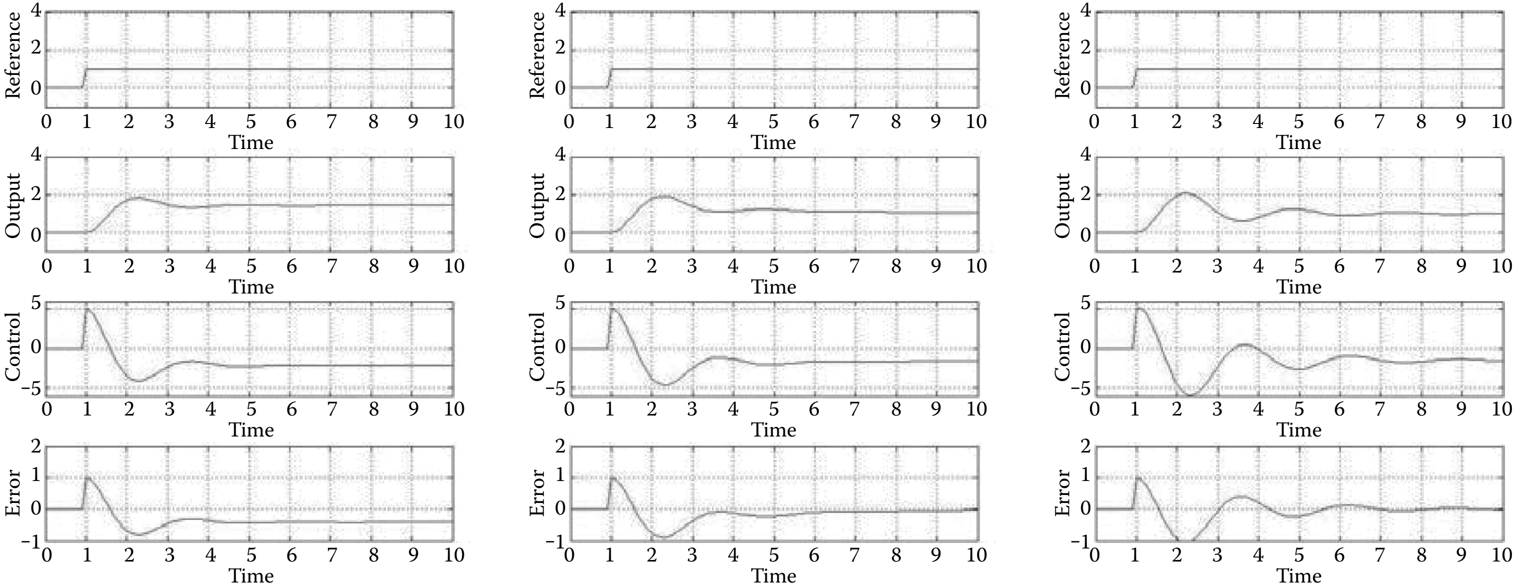


FIGURE 4.10
Performance of a PI control of K_P of 5.0 with K_I of 0.0, 1.0, and 3.0 to $G(s)$.

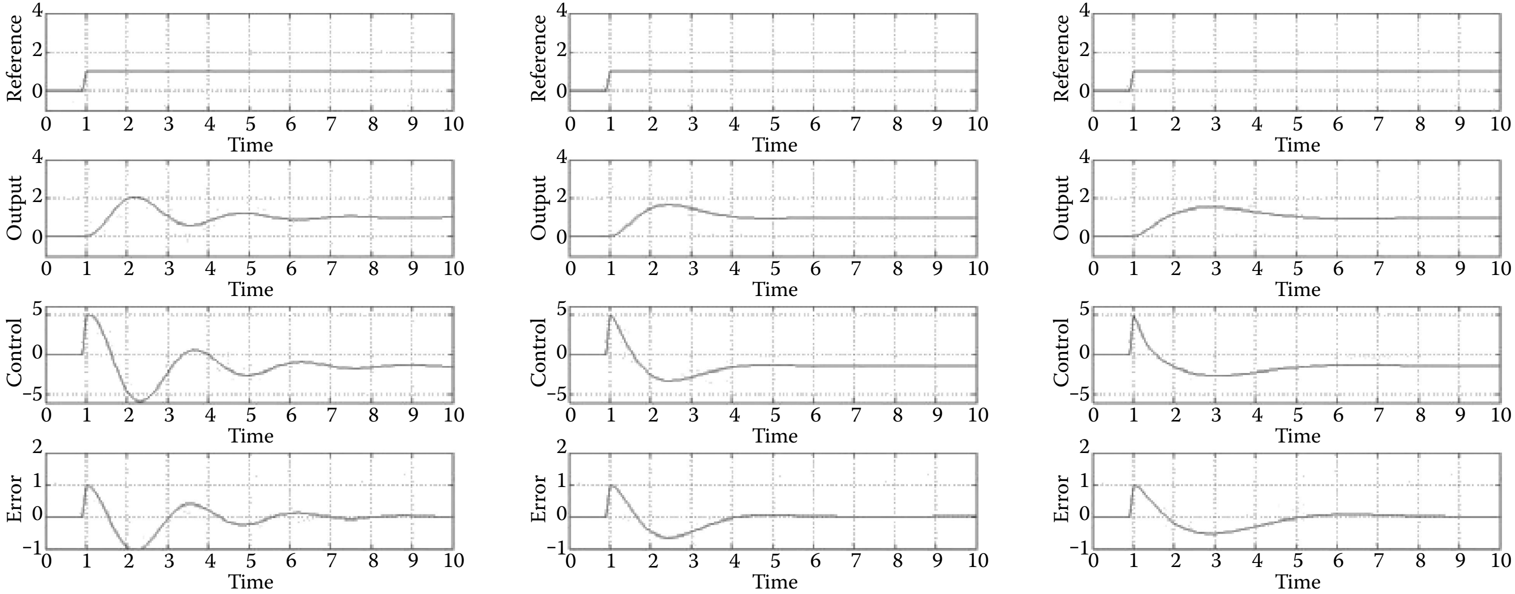


FIGURE 4.11
Performance of a PID control of K_P of 5.0 and K_I of 3.0 with K_D of 0.0, 1.0, and 2.0 to $G(s)$.

TABLE 4.1

Effects of PID Gains on System Response Speed and Stability

Gain	**Response Speed**	**Stability**
Increasing K_P	⇧	⇩
Increasing K_I (decreasing T_I)	⇧	⇩
Increasing K_D (increasing T_D)	⇧	⇩

The robustness of the PID controller is also related to the demand for speed. Robustness means the sensitivity to variations in the process dynamics. Controllers that are tuned to give fast control are usually more sensitive to variations in the process than controllers that are more conservative in their settings.

Table 4.1 presents a guide to the effect of each PID gain to the speed and stability of the system and how stability and speed change when the gains are changed. Table 4.1 only contains rules of thumb, and there are exceptions. For example, an increased gain often results in more stable control for low-order processes, such as those involving liquid level control.

Tuning of the PID gains is usually performed in the order of proportional, integral, and derivative. Initially, the integral and derivative parts are effectively disconnected by either switching them off or setting T_I very high and setting T_D very small. After K_P has been adjusted according to the guide in Table 4.1 such that the performance approaches the desired one, T_I is reduced to a suitable setting for the integral part.

Referring to Table 4.1, this will result in a reduction in stability, which, in turn, means that the gain K_P has to be reduced. One significant exception to this is in liquid flow control. Liquid flow control loops are very fast and quite often tend to be very noisy. As a result, integral action is often added to the feedback controller in liquid flow control loops to provide a dampening or filtering action for the loop. The advantage of eliminating offset is still present in these cases, but it is not the principal motivating factor.

When the proportional–integral controller exhibits satisfactory performance, the adjustment for T_D may begin. Increase in T_D will typically result in an improvement in stability, which, in turn, means that the gain K_P may be further increased and T_D further reduced. The derivative part, according to Table 4.1, produces both faster and more stable control when T_D is increased. This is only true up to a certain limit and if the signal is sufficiently free of noise. Rising T_D above such a limit will result in reduced stability in control. As mentioned previously, the function of the derivative part is to estimate the change in the control at time T_D ahead. This estimation will naturally be poor for large values of T_D. The reasoning above has also been made without taking into account noise or other disturbances. The noise is amplified to a greater extent when T_D increases. It is thus often the noise that sets the upper limit for the magnitude of T_D. In some cases, the noise level can be so high, despite filtering, that one may be unwilling to use the derivative part because it gives such a bad control signal. A proportional–integral controller would then be preferred with its smoother control signal even at the cost of poorer control performance.

A systematic tuning method called the Ziegler–Nichols* is often employed to quickly obtain a set of gain that broadly meets the requirements. The Ziegler–Nichols method uses information from the step response or the frequency response of the system for tuning the PID controller using a set of formula as follows:

---

* The method was invented and developed by two engineers at the Taylor Instrument Company, John Ziegler and Nathaniel Nichols, in the 1940s.

$$K_P = \frac{1}{KT_D\tilde{T}\omega_0^2} \tag{4.21}$$

$$T_I = \frac{2\zeta}{\omega_0} \tag{4.22}$$

$$T_I T_D = \frac{1}{\omega_0^2}\,. \tag{4.23}$$

where the damping ratio ζ and the system's natural frequency ω_0 are identified from the system's step response, and the desired closed-loop time constant $\tilde{T}$ is determined according to the application.

Control saturation, caused by the limited range of control action, can lead to an integrator windup phenomenon, and takes place with the presence of integral control as follows: When the change of reference signal is high, the integral control causes the control signal to increase, which can go up to its upper limit (u_{max}), and yet is insufficient to eliminate the control error. When the reference is later changed to a level within range of the control signal, the control signal will remain at its upper limit for a longer time before it unwinds in accordance with the errors. The windup duration is longer in systems with longer time constants or large process gains because the corresponding small proportional and integral action prolongs the reversal of controller output. Figure 4.12 shows the control signal, the controlled variable (process variable), and the reference (set point) in a case in which the control signal is limited. After the first change in set point, the control signal increases up to its upper limit u_{max}. This control signal is not large enough to eliminate the control error. The integral part of the PID controller is proportional to the area under the control error curve. This area is marked in Figure 4.12. Hence, the integral part will continue to rise because the control signal is unable to eliminate the control error.

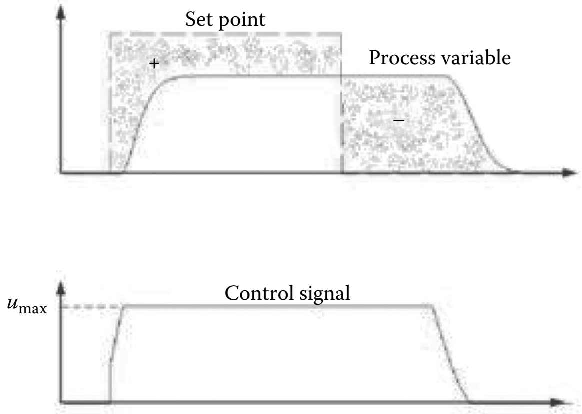


FIGURE 4.12
Integrator windup.

Figure 4.12 also shows what happens when, after a certain time, the reference is changed to a lower level at which the controller is able to eliminate the control error. Because the integral part was allowed to rise and reach a high level while the control signal was limited, the control signal remains at its limit for a longer time before it *unwinds* in accordance with the negative errors.

This problem is known as integrator or reset windup, and the consequence is a response with a large overshoot and long settling time. Integrator windup commonly occurs in override, surge, batch, and pH control loops. The duration of the windup is longer for those systems with longer time constants and large process gains because the corresponding small proportional and integral action prolongs the reversal of controller output.

There are several ways to avoid integrator windup. The simplest way of overcoming the problem is to stop updating the integral part when the control signal is limited. For this to be possible, the controller naturally has to know what the limits are. In Figure 4.13, this point in time corresponds to point B. The question now is determining at what point it is enabled again. Point C is obviously far too late (although it is still better than point D in the unprotected controller). A common solution is to desaturate the integral term at the point at which the rate of increase of the integral action equals the rate of decrease of the proportional and derivative terms. This occurs when the slope of the PID output is zero, that is, when

$$e = -T_\mathrm{i}\left(\frac{\mathrm{d}e}{\mathrm{d}t} + T_\mathrm{d}\frac{\mathrm{d}^2 e}{\mathrm{d}t^2}\right). \tag{4.24}$$

This brings the controller out of saturation at the earliest possible moment, but this can, in some cases, be too soon, leading to an unnecessarily damped response. Most controllers are equipped with methods, in one form or another, for avoiding integrator windup.

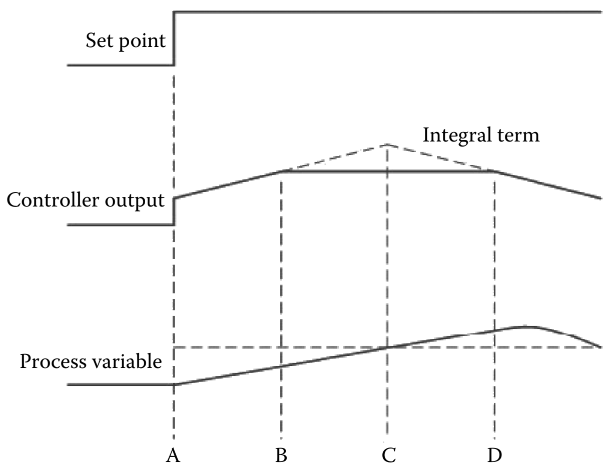


FIGURE 4.13
Effect of integrator windup.

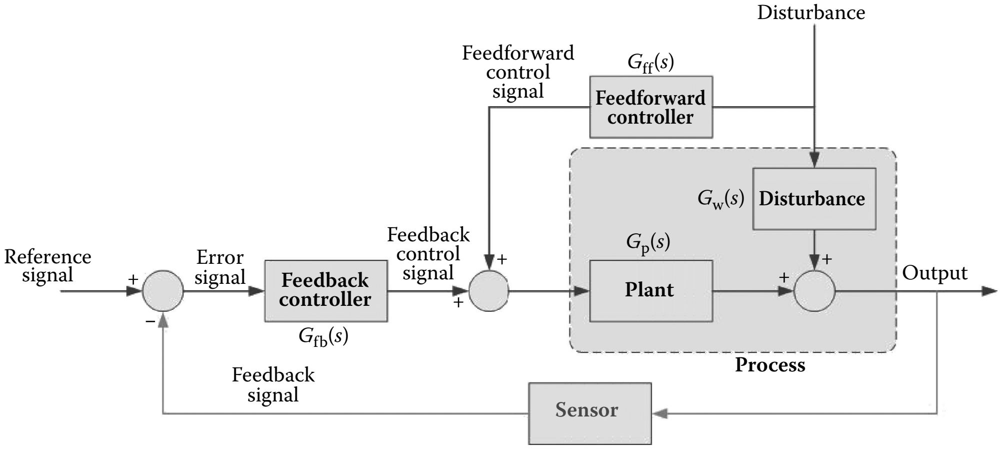

Disturbance
Feedforward control signal
$G_{ff}(s)$
Feedforward controller
$G_w(s)$
Disturbance
Feedback control signal
$G_p(s)$
Reference signal
Error signal
Feedback controller
Plant
Output
$G_{fb}(s)$
Process
Feedback signal
Sensor
+
−
+
+
+
+

FIGURE 4.14
Feedforward–feedback control structure.

4.3.2 Feedforward Control

Feedforward control is used to reject a measurable disturbance before it affects the output of the system. Feedforward control is typically used in tandem with a feedback control; in this configuration, feedforward is used solely to reject the major disturbance, and the feedback is used to track the reference value as well as rejecting other possible disturbances. The feedforward plus feedback configuration is also referred to as a two-degree-of-freedom (2DOF) control. To achieve maximum advantage and outweigh the cost of implementation, it is therefore important that the disturbance to be rejected is a major disturbance and that a reasonably good model of the disturbance is available. Figure 4.14 presents the classical feedforward–feedback control structure.

In the simplest terms, based on the design, feedforward control is started by defining the transfer function of the output with respect to the disturbance as follows:

$$H(s) = \frac{G_w(s) - G_p(s)G_{ff}(s)}{1 + G_{fb}(s)G_p(s)} \tag{4.25}$$

Eliminating the disturbance requires $G_w(s) - G_p(s)G_{ff}(s) = 0$. For a typical system in which the dead time and relative order of $G_w(s)$ are greater than those of $G_p(s)$ and in which $G_p(s)$ has no right half plane zeros, the design of the feedforward control is such that

$$G_{ff}(s) = G_p^{-1}(s)G_w(s) \tag{4.26}$$

4.3.3 Servo Control Structures

A generally applicable servo control structure used for servo control is shown in Figure 4.15. The structure is shown with reference to a position control problem for illustration although it can just as efficiently be adopted for other controlled variables, such as speed, force, and pressure.

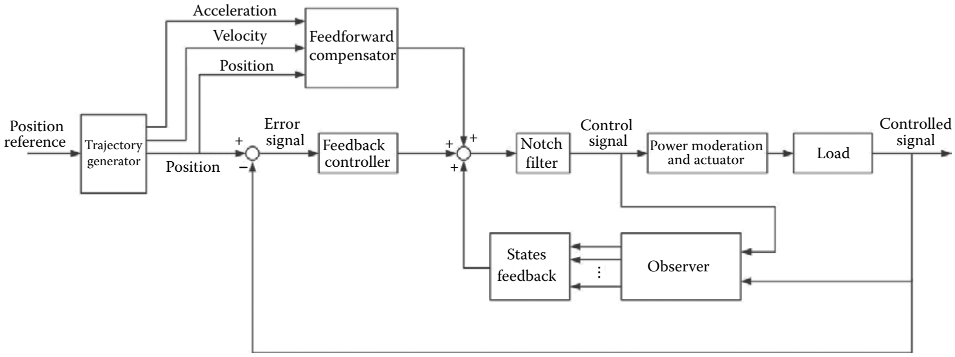

Acceleration
Velocity
Position
Feedforward compensator
Position reference
Trajectory generator
Position
Error signal
+
−
Feedback controller
+
+
+
Notch filter
Control signal
Power moderation and actuator
Load
Controlled signal
States feedback
Observer

FIGURE 4.15
Servo control structure.

The feedback controller and the feedforward compensator or controller blocks consist of the control schemes as described in Sections 4.3.1 and 4.3.2, respectively. Although these controllers are able to drive the system to follow the desired trajectory (which can be generated by a trajectory generator based on certain references), additional blocks are essential, too.

The notch filter is a filter that nearly nulls the response at very specific frequencies. A notch filter is useful in many applications with specific undesired frequencies, such as in systems with high interference from an AC power line. In such a system, a notch filter to nullify the power line frequency (typically either 50 Hz or 60 Hz) is employed.

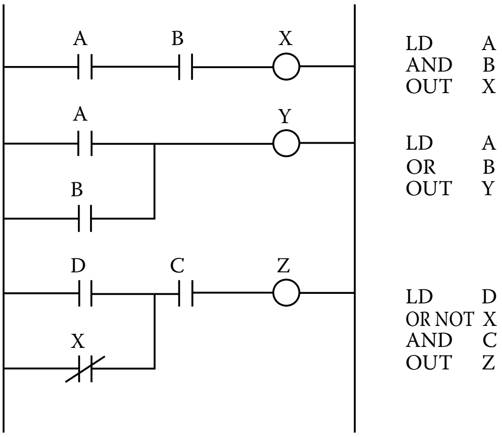
The control signal out of the notch filter (or out of the controller when a notch filter is omitted) is used to drive actuators after being moderated to a suitable signal. As illustrated through examples in Section 4.6, the power moderators condition a typically low-magnitude control signal into a proportionately high-magnitude (and sometimes with suitable frequency and phase, too) driving signal to power the actuators and, in turn, to drive the load.

Control feedback necessitates the measurement of all the states of a system to be fed back to the controller, and yet it is often not feasible to measure all the states due to practicality, cost, or other reasons. This is when an observer is employed to estimate all of the states of the system from a limited number of actual measurements. The efficacy of the observer depends on the accuracy of the observer model in estimating the states of the system.

4.3.4 Programmable Logic Controllers

The programmable logic controller (PLC) is a common form of control platform for mechatronic systems that is based on logic rules. It is dedicated and hardy piece of equipment, which is used in almost all industries, including automobile production lines, brewing, water treatment, assembly lines, etc. The PLC is realized via any of the five languages as regulated in the IEC 61131-3 standard as follows:

1. Ladder diagrams
2. Instruction lists
3. Structured text
4. Sequential function charts
5. Function block diagrams



```
LD      A
AND     B
OUT     X

LD      A
OR      B
OUT     Y

LD      D
OR NOT  X
AND     C
OUT     Z
```

FIGURE 4.16
A ladder diagram and an instruction list executing the same command.

These five languages are different in form but the same in terms of substance and may be used concurrently. A ladder diagram is rather similar to a relay ladder diagram or program. It implements logic via a combination of switches, which are either normally open or normally closed. An instruction list is a low-level language that has a structure similar to a simple machine assembler and is therefore ideal for solving straightforward problems with which there are few decision points and with which there are a limited number of changes in program execution flow. A structured text is a high-level, procedural language that can be used for programming a wide range of industrial applications. It has facilities for conditional evaluation of statements, for repeating sections of code, and for calling functions and function blocks. A sequential function chart is a graphical language for dictating sequences, hence making it possible to provide alternative sequences to be selected using divergent paths. A function block diagram is graphics-based software with building blocks to construct the entire control system.

Figure 4.16 shows an example of using two of these PLC languages: a ladder diagram (on the left) and an instruction list (on the right).

4.4 Implementation of a Computer Control

Most controllers today are implemented in computer-based distributed control systems (DCSs). The advantage of DCS implementation includes the possibility of incorporating more complex functions in the controller, such as automatic tuning, alarm handling, filtering, digital control, etc. The parameter values can also be tuned more accurately because the user settings will be precisely the true control settings, unlike analog control with which they may be different due to aging of analog components. The disadvantage, however, is due to the inherent discrete nature of a digital system. Between consecutive samples, the controller gets no information on the intermediate value of the analog signal. This is not a problem so long as the time interval between the samplings is much shorter than the time constant of the controlled process, but this is not always the case. Sampling is also effectively equivalent to introducing an additional dead time into the feedback loop.

A sudden disturbance is only detected by the control system after an average time of half a sampling period. An understanding of this phenomena and the importance of selecting an appropriate sampling interval to avoid issues, such as aliasing (as presented in Figure 4.17), is therefore crucial to the performance of a control system.

Implementing a continuous time controller using a DCS requires an approximation with difference equations. In the case of a PID controller, as explained in the previous section, the approximation is given in the following.

The continuous time proportional term of

$$u_P(t) = K_c e(t) \tag{4.27}$$

is implemented simply by replacing the continuous variables with their samples as follows:

$$u_P(\mathrm{kh}) = K_c e(\mathrm{kh}) \tag{4.28}$$

where h is the sampling interval and kh denotes the k^{th} sampling instance.

The continuous time integral term of

$$u_I(t) = \frac{K_c}{T_I} \int_0^t e(\tau)\, d\tau \tag{4.29}$$

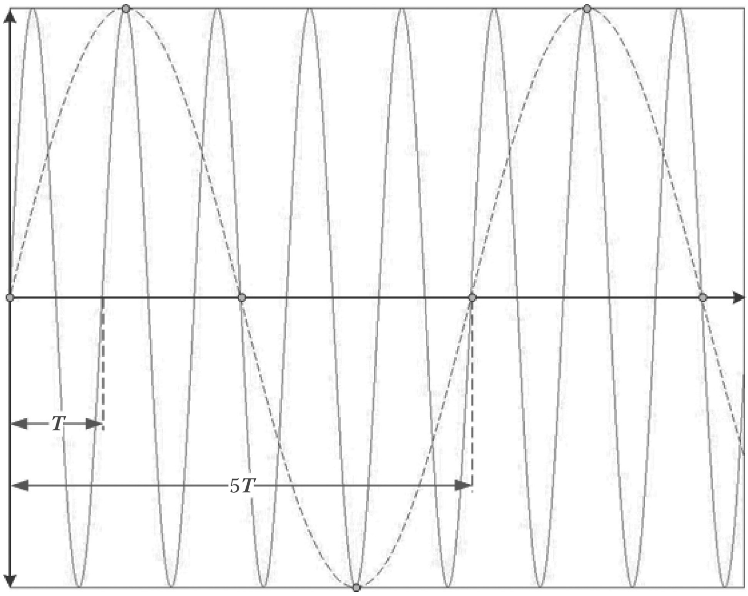


FIGURE 4.17
Aliasing with which a signal with a period of ***T*** is mistaken as 5***T*** due to the sampling interval of 1.25***T***.

where $e(t) = r(t) - y(t)$. In discrete time, the integral may be approximated by the area under the error curve from $t = 0$ to $t = \mathrm{kh}$ iteratively as follows:

$$u_{\mathrm{I}}(\mathrm{kh}) = u_{\mathrm{I}}((k-1)h) + \frac{K_{\mathrm{C}}}{T_{\mathrm{I}}} A \tag{4.30}$$

where A is the area under the error curve from $t = (k - 1)h$ to kh. As shown in Figure 4.18, A may be approximated in various ways.

The continuous time derivative term with filtering is as follows:

$$\frac{T_{\mathrm{D}}}{N} \frac{\mathrm{d}u_{\mathrm{D}}(t)}{\mathrm{d}t} + u_{\mathrm{D}}(t) = K_{\mathrm{C}} T_{\mathrm{D}} \frac{\mathrm{d}e}{\mathrm{d}t} \tag{4.31}$$

Integrating this equation, it follows that:

$$\begin{aligned} \frac{T_{\mathrm{D}}}{N} u_{\mathrm{D}}(\mathrm{kh}) + \int_{(\mathrm{k}-1)\mathrm{h}}^{\mathrm{kh}} u_{\mathrm{D}}(t)\,\mathrm{d}t &= \frac{T_{\mathrm{D}}}{N} u_{\mathrm{D}}((k-1)h) \\ &+ K_{\mathrm{C}} T_{\mathrm{D}}(e(\mathrm{kh}) - e((k-1)h)) \end{aligned} \tag{4.32}$$

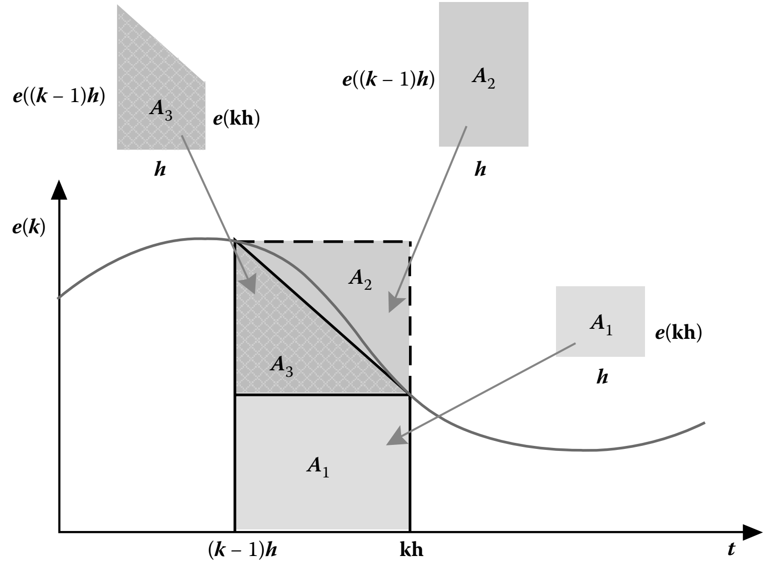


FIGURE 4.18
Discretization of the area under the error curve either as $A_1 = he(\mathbf{kh})$, $A_2 = he((k-1)h)$, or $A_3 = h\frac{e(k-1)h + e(\mathbf{kh})}{2}$.

Denoting $A = \int_{(k-1)h}^{kh} u_D \, dt$, A may be computed in various ways similar to those presented in Figure 4.18. In discrete time, the derivative part is discretized according to one of the approximations of A such that the corresponding difference equation for $u_D(kh)$ is

$$u_D(kh) = \frac{T_D}{hN + T_D} u_D((k-1)h) + \frac{K_C T_D N}{hN + T_D}(e(kh) - e((k-1)h)), \text{ or} \tag{4.33}$$

$$u_D(kh) = \frac{T_D - hN}{T_D} u_D((k-1)h) + \frac{K_C T_D N}{T_D}(e(kh) - e((k-1)h)), \text{ or} \tag{4.34}$$

$$u_D(kh) = \frac{2T_D - hN}{hN + 2T_D} u_D((k-1)h) + \frac{2K_C T_D N}{hN + 2T_D}(e(kh) - e((k-1)h)). \tag{4.35}$$

These discrete approximations will be stable if the sampling interval h is sufficiently small.

The inputs and outputs of a digital control system can be of various kinds. They can be direct analog signals, direct digital signals, or a bus interface. Modern control systems are designed to be modular in nature so that, apart from the core control module, the number and types of I/O modules can be optimized to the requirements.

4.5 Challenges in Control of Mechatronic Systems

Control systems have to deal with nonlinearities, which always exist in different elements of the system, be it in the sensors, actuators, or even the controller. Such nonlinearities can take a few possible unique forms, which also require unique techniques to address the issues.

4.5.1 Friction

Friction manifests itself prominently in moving mechanisms of any control system. In a hydraulic drive, for example, friction in the control valves is one of the major reasons for poor performance, causing stick–slip motion and oscillations. Friction varies due to time, location, and even temperature as higher temperatures cause material expansion and, hence higher frictional force. The presence of friction is often exacerbated by implementation practicality, for instance, when packing boxes of valves requires tightening to avoid leakage over time but with an adverse effect of increasing friction.

In a different system, such as an electric drive, friction occurs at the interface of moving mechanisms where the electric power is transformed into mechanical motion. Friction in this instance is typically categorized into either static or dynamic friction. Static friction, including stiction, kinetic force, viscous force, and the Stribeck effect, is a function of steady state velocity. Dynamic friction, including presliding displacement, varying breakaway force, and frictional lag, is conditional upon changing velocity. Many empirical friction

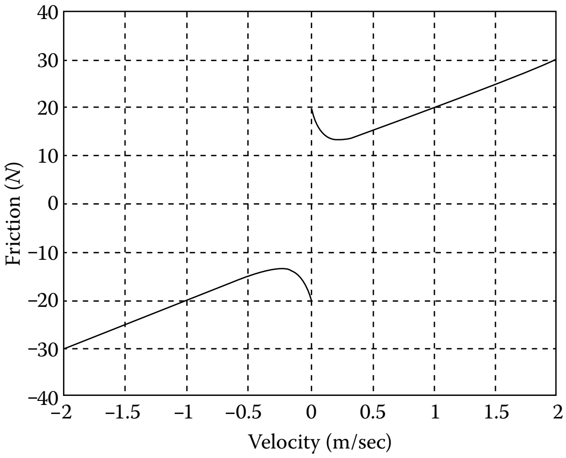


FIGURE 4.19
Tustin friction model.

models have been developed that attempt to capture specific components of observed friction behavior, one of which is a friction model as presented in Figure 4.19.

An understanding of friction phenomena is important in the control of mechatronic systems. The symptom of friction, such as oscillations, is the same as other phenomena such as controller overtuning, and is easily mistaken for such. Most adaptive controllers behave in the same oscillating way, too, when in the process of adaptation. Distinguishing whether the phenomena is caused by bad tuning or by the presence of friction in the actuation is therefore important.

Remedying the effect of friction requires an analysis of its cause. Friction arising from mechanical transmission, for example, can be reduced by using more efficient transmission mechanisms, such as air and magnetic bearings.

4.5.2 Force Ripples

Force ripples occur in almost all types of electric motors in which a ferromagnetic core is used for the windings. The two primary components of force ripple are cogging force and reluctance force. The cogging force arises as a result of the mutual attraction between the magnets and iron cores of the rotor and exists even in the absence of any winding current. It exhibits a periodic relationship with respect to the position of the rotor relative to the magnets. Cogging manifests itself by the tendency of the rotor to align in a number of preferred positions regardless of excitation states. Reluctance force, on the other hand, is due to the variation of the self-inductance of the windings with respect to the relative position between the rotor and the magnets. Reluctance force also has a periodic relationship with the rotor and the magnet's relative position.

Force ripple has significant effects on the position accuracy achievable and may also cause oscillations and yield stability problems, particularly at low velocities or light loads (low momentum). Figure 4.20 shows the real-time displacement ramp response of a tubular type of permanent magnet linear motor in which the force ripples are especially evident over the portion with constant velocity.

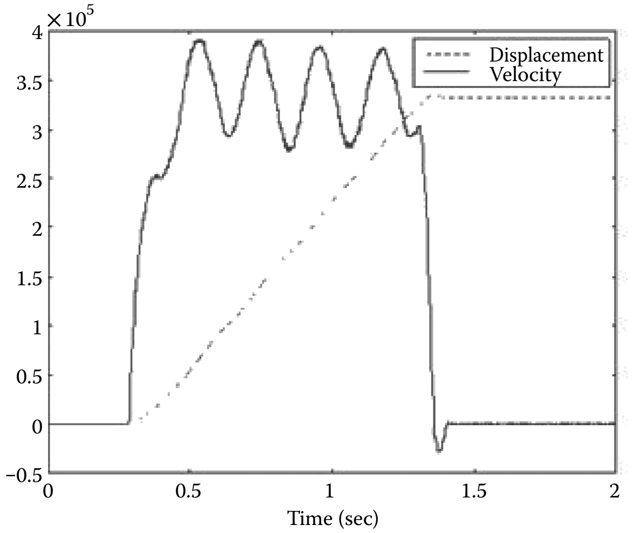


FIGURE 4.20
Force ripple.

4.5.3 Hysteresis and Backlash

Poorly machined gears or loose-fitting linkages exhibit backlash. This manifests itself as a differing input/output relationship according to the direction of movement of the input shaft. In a similar manner, hysteresis can be a source of problems and often manifests itself as a dither about the set point as the control system hunts in the dead band. It can also be self-reinforcing as dither will lead to more wear and more backlashes. Backlash and hysteresis can be visualized as in Figure 4.21. Backlash and hysteresis is best handled by

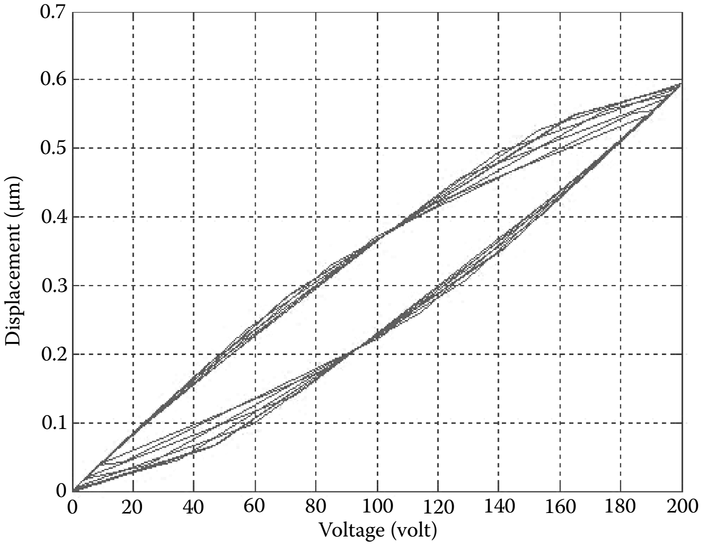


FIGURE 4.21
Backlash/hysteresis model.

design and careful manufacturing techniques that provide preload, such as spring-loaded gears and pretensioned linkages.

4.5.4 Saturation

The signals in control loops are always limited. The measuring sensors have their own working ranges. If the measured quantity falls outside the measurement range of the sensor, the signal will be limited. In the same way, the control signal is also limited as shown in Figure 4.22.

On the actuator side, saturation also occurs. A valve, for example, has its working range between fully closed and fully opened. The speed of a DC motor is limited to prevent centrifugal forces from damaging the motor and couplings. The side effect of saturation is that it effectively reduces the gain at high amplitudes and hence slows the system response to disturbances.

Limitation of the control signal can cause special problems if the controller is not informed when this is occurring. In controllers, the limitations on the control signals are often specified, and the range should be adhered to as far as possible. Operation outside of the control signal range may risk the controller to a phenomenon known as integral windup.

4.5.5 Dead Zone

A dead zone (presented in Figure 4.23) may arise inadvertently due to friction, or it may deliberately prevent unnecessary response to small errors. A typical application is a level control system in a hydraulic surge tank in which the level is allowed to vary within limits without corrective action being taken, and it is undesirable for the control system to attempt to correct for ripples and hydraulic resonance.

4.5.6 Reference Signal Changes

In servo control, the reference signal changes according to the desired motion profile. This reference disturbance is preplanned and initiated by the operator or controller. Thus, it is

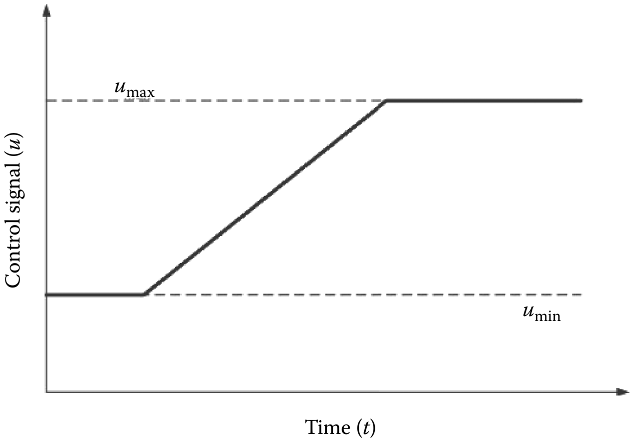


FIGURE 4.22
Saturation model.

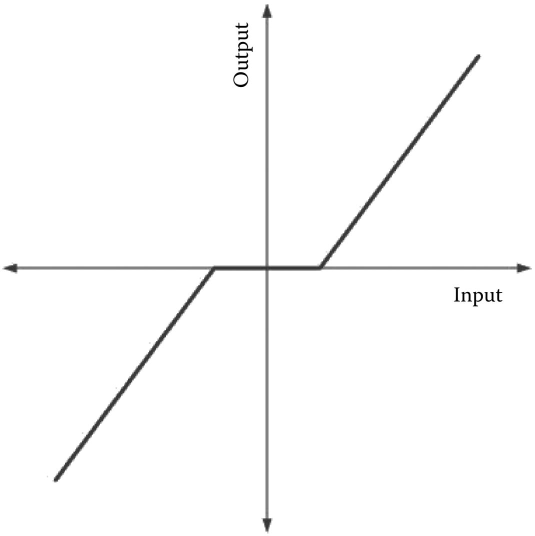


FIGURE 4.23
Dead zone model.

one of those known disturbances that the controller will be typically designed and well positioned to address.

4.5.7 Low-Frequency Drift

Low-frequency disturbances, such as slow drift in the load disturbance, are compensated for using the controller. These may arise due to load change or changes in the drive mechanisms. In a PID controller, the integral part ensures that a high gain is available in these frequency ranges and therefore effectively eliminates these disturbances.

Disturbances that lie in the same frequency range as the system's own dynamics are probably the most difficult to handle. These oscillations cannot be easily filtered out without filtering out other useful information. In addition, the controller also has difficulty in rejecting these disturbances. If possible, these disturbances should be eliminated from the feedback loop. The best way, if possible, is to eliminate them at the source through proper design of the system and its components. If this is not possible, then feedforward from the source may be an effective method to eliminate their effects before they affect control performance.

4.5.8 High-Frequency Noise

High-frequency disturbances at the input to the system are usually controlled effectively by the system itself because nearly all physical systems have the characteristics of a low-pass filter. The high-frequency components present in the controlled variables are therefore usually generated in the sensor or the lead from the sensor. They can also arise due to switching power moderators, such as a pulse-width moderator (PWM). These high-frequency disturbances or noise should not be addressed at the controller. In order to avoid wear and tear on the actuators, the noise should be filtered out before the controlled variable is transmitted to the control algorithm.

If the noise level is so high that the normal filtering in the controller is not sufficient, the controlled signal should be filtered further. In most modern controllers, the facility is therefore provided for passing the controlled signal through a low-pass filter. The filter bandwidth should be chosen to be wide enough so that the filter does not affect the lower-frequency signals in the same band as the controller. The choice of filter bandwidth is a compromise between speed and stability. With hard filtering (long filter time constant), smooth signals may be obtained with little wear on the valves and actuators as a result. On the other hand, the consequence is a slower handling of load disturbances as the effect of these is also filtered.

4.5.9 Incorporating and Addressing Nonlinear Dynamics

Nonlinearities are incorporated into the dynamics of a mechatronic control system for modeling and designing of the controller. When, for example, the friction, force ripple, and hysteresis of a control system have been determined to be significant, the dynamics of the system are modeled as follows:

$$\ddot{x} = -\frac{K_1}{M}\dot{x} + \frac{K_2}{M}u - \frac{1}{M}(F_{\text{load}} + F_{\text{ripple}} + F_{\text{friction}} + F_{\text{hysteresis}}) \tag{4.36}$$

Such a model can be simplified by grouping all nonlinearities into one common term, such as $F_1^*(x, \dot{x})$, to represent the nonlinear dynamical effects due to force ripple, friction, and other unaccounted dynamics collectively. The system can thus be alternatively described by

$$\ddot{x} = -\frac{K_1}{M}\dot{x} + \frac{K_2}{M}u - \frac{1}{M}F_{\text{load}} + F_1^*(x, \dot{x}) \tag{4.37}$$

When the load is also a function of x or $\dot{x}$, one can go a step further to group the load and the nonlinearities as one term as follows:

$$\frac{K_2}{M}f(x, \dot{x}) = -\frac{1}{M}F_{\text{load}} + F_1^*(x, \dot{x}) \tag{4.38}$$

such that

$$\ddot{x} = -\frac{K_1}{M}\dot{x} + \frac{K_2}{M}u + \frac{K_2}{M}f(x, \dot{x}) \tag{4.39}$$

Addressing the nonlinear dynamics is typically aimed at minimizing the error. For a typical system in which the first-order tracking error* is to be minimized, the tracking error is defined as

$$e = x_{\text{d}} - x \tag{4.40}$$

where x_{d} is the desired value of x.

---

* A first-order tracking error can be thought of in the perspective of a system whose variables follow a set of desired values. Examples include a track following vehicle.

As the aim is to minimize the tracking error, the equation is converted into an error equation of

$$\ddot{e} = -\frac{K_1}{M}\dot{e} - \frac{K_2}{M}u - \frac{K_2}{M}f(x,\dot{x}) + \frac{K_2}{M}\left(\frac{M}{K_2}\ddot{x}_d + \frac{K_1}{K_2}\dot{x}_d\right). \tag{4.41}$$

Because

$$\frac{d}{dt}\int_0^t e(t)\,dt = e, \tag{4.42}$$

the system state variables are assigned as follows:

$$x_1 = \int_0^t e(t)\,dt = e \tag{4.43}$$

$$x_2 = e \tag{4.44}$$

$$x_3 = \dot{e}. \tag{4.45}$$

This is where state space form (as discussed in Section 4.2) is employed by taking the following definition of x:

$$x = [x_1\ x_2\ x_3]^T \tag{4.46}$$

such that the equation can be restated as follows:

$$\dot{x} = Ax + Bu + Bf(x,\dot{x}) + B\left(-\frac{M}{K_2}\ddot{x}_d - \frac{K_1}{K_2}\dot{x}_d\right), \tag{4.47}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{K_1}{M} \end{bmatrix}, \tag{4.48}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -\frac{K_2}{M} \end{bmatrix}. \tag{4.49}$$

To compensate dynamically for the effect of unmodeled mechanical behaviors on the quality of the straight motion, an internal model control (IMC) scheme can be added. This consists of introducing into the control a variable characterizing the difference between the output of the process and modeling. Its value is then converted into an image of the extra torque required to compensate unmodeled loads and is added to the motor's command calculated by, for example, the PID controller. The control scheme is presented in Figure 4.24.

Block M represents the model of the process and M^{-1} its inverse. Moreover, the calculus of the modeling error is achieved at the acceleration level of the control in order to avoid the presence of integral terms in the model M. It means that the modeling error is considered to be the difference between the variables predicted by the model and the one measured. The real variable is obtained by a double numerical derivation of the information provided by the linear incremental encoder. A low-pass numerical filter can be used to reduce the amount of high-frequency noise introduced by the derivations and focus the action of the IMC loop on very low-frequency disturbances. Indeed, if the modeling error calculated at the sample time n contains a lot of numerical noise when used to modify the value of the motor's command at the time $n + 1$, it acts as a powerful noise generator, which goes against the initial purpose of the IMC. The order of the filter was limited to reduce dephasing. The very low cutoff frequency of the filter does not affect the natural frequency of the control as the IMC loop acts independently from the PID control, which determines the dynamics of the servo system (Figure 4.25).

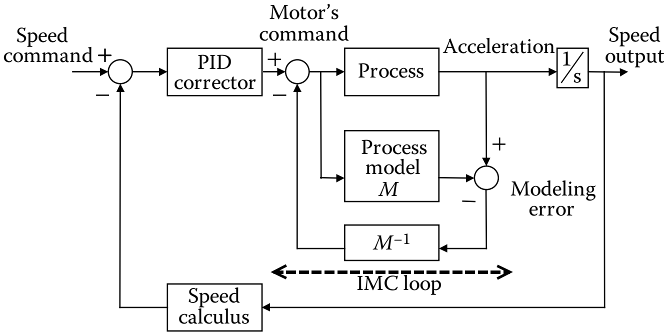


FIGURE 4.24
Internal model control (IMC) applied to the PID speed control.

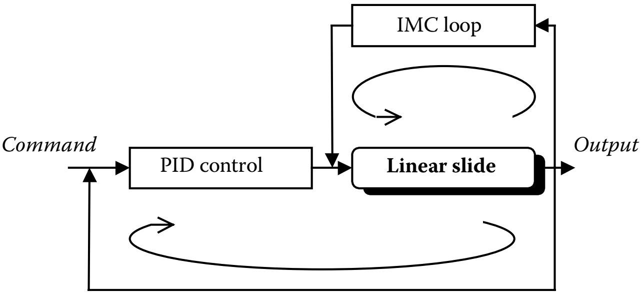


FIGURE 4.25
Separate action of the IMC loop and PID servo control.

4.6 Application Examples

4.6.1 Flight Simulators

A flight simulator is designed such that it will emulate the actual situations of an airplane on the ground and in the sky as closely as possible to match various plausible scenarios that an aspiring pilot is likely to encounter in real life. Through the simulator, the trainee or even a professional pilot can experience the feel of flying an airplane at hardly a fraction of the risk incurred.

A flight simulator generates different motion profiles through a multiple-degree-of-freedom motion platform based on the kinematics and dynamics of an actual plane and according to the pilot commands, current plane parameters, and possible exceptional situations.

A flight simulator can be decomposed into several key functions, working in synchronization.

In the cockpit, the pilot will be able to know the current flight parameters through arrays of display panels, meters, and alarms. These devices essentially display important measurements, such as the current plane orientation, the current flight profile, and wind conditions. They will also show the operational status of key parts of the airplane, such as wings, ailerons, and tails, and will sound or display an alarm when faults and malfunctions are detected. This information is necessary for the pilot to make informed decisions on any necessary corrective actions. This measurement function is accomplished via a set of sensors and instrumentation circuits, which will collectively collect the data and convert them to a form that is amenable for display and control purposes.

A framework of actuators is installed right below the cockpit to generate and deliver the appropriate motion profile of the entire platform according to the pilot's commands as well as the effects of varying simulated weather conditions and turbulences. The cockpit is the load that is driven by these actuators, which can be pneumatic, hydraulic, or electromagnetic actuators.

The actuators are capable of generating multiple-degree-of-freedom motion to the cockpit to yield the same sense of flight to the pilot as in a real airplane. Each actuator therefore needs to be controlled to such precision as to create an overall effect that will emulate the real situation as closely as possible. In addition, the interaction and coupling between the individual actuators within the framework has to be adequately addressed, too.

Power moderation is the intermediary function to match the pilot or control command signal to the final actuation signal for the actuator, both in form and energy level. The specific moderation varies considerably, depending on the overall system design. The moderation process may include amplification, conversion, or switching, depending on the overall system design and the type and operational principle of the actuators used.

The role of power moderation is to enable adequate communication among different components of the servo control system, considering that different components at different levels operate at different magnitude order. Suppose the flight simulator uses DC electric motors for actuation, requiring a few hundreds of volts. Measurement signals from sensors, on the other hand, are usually in lower energy forms on the order of millivolts (mV). Power moderation may then involve signal amplification to amplify the measurement signal to a few volts to be processed by a computer or be displayed on a meter and then to the level of hundreds of volts to drive the motors. Amplification does not only apply to electrical signals; there are also pneumatic and hydraulic amplifiers.

For different types of actuators, such as pneumatic, hydraulic, or AC motors, other kinds of moderation may be necessary to convert the control signal to an alternate signal form, which is needed to operate these actuators. AC motors, for example, may also require phase matching for their operation. Apart from amplification, other common moderation includes current-to-voltage conversion, current-to-pressure conversion, signal switching, and signal chopping.

The control function in a servo control system is the core function, which helps to integrate all other functions. Essentially, it generates and transmits the specific motion commands to the power moderator.

From the cockpit of the flight simulator, the pilot can give instructions and commands via the control panel on target flight parameters to achieve, such as speed and flight path. The motion commands can be directly transmitted to the power moderator so that the pilot directly controls the actuators. Alternatively, the motion command can serve as the reference signal or set point for an automatic control system. In this case, the trainee pilot may only need to specify the target speed and flight path, and the automatic control system uses the specifications to manipulate the actuators based on the feedback measurements. The first case is called open loop control, and the latter case is called closed loop control.

Apart from executing pilot commands for standard actions related to takeoff, landing, turning, and maneuvering, the control system also needs to respond adequately to exceptional conditions, such as head wind, turbulence, storms, heavy rain, or even malfunction of some parts of the simulator. From a control system perspective, these are the disturbances that may disrupt the otherwise well-controlled systems. Thus, apart from tracking and executing commands, the control system in a flight simulator has to remain robust to the disturbances a plane may experience.

These various components are now ready to be fitted in place together to result in a complete system. Figure 4.26 presents the integration of these components in a simple control system.

A reference signal or command (represented by x_r in Figure 4.26) is in put into the system as the desired motion profile to be achieved. This command can be an input from the pilot or, in the case of an autopilot system, one from a higher-level supervisory control system. The present output of the system is the controlled motion variable, and it can be measured by sensors to yield the feedback signal. This feedback signal is then compared to the reference signal, the difference of which is the error signal. Based on the error signal, the controller will output a necessary control signal to the actuator, via the power moderator, so as to drive the load according to the desired motion profile.

In addition, disturbance signals are also highlighted in Figure 4.26 (represented by x_d), representing the effects of extraneous signals seeping into the loop. The environment alone

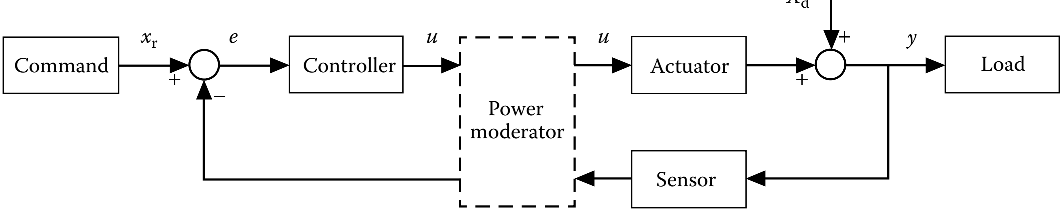


FIGURE 4.26
A flight simulator.

will expose the system to various conditions, which can be unpredictable. Disturbances, if not adequately compensated for, can cause the controlled variable to deviate from the reference signal. For example, due to tail wind, the plane can be accelerated beyond the desired profile. With a feedback controller, the control action will have to be reduced to bring the plane back on track.

Collectively, the overall system integrating these functions is also often referred to as a servo drive. It will be a hydraulic servo drive if hydraulic pressure is used as the mean for actuation or an electric servo drive if an electromotive force is used. However, it is important to clarify that this is not necessarily a universal definition. To some manufacturers, servo control may exclude the actuation function.

4.6.2 Piezoelectric Control System for Biomedical Application

This section discusses the design of an intra-cytoplasmic sperm injection (ICSI) installation. It involves the design of the actuation system, sensing system, and control system, constructing a typical mechatronic system.

ICSI is a laboratory procedure developed to help infertile couples undergoing *in vitro* fertilization (fertilization on the dish) due to male factor infertility. It involves the injection of a single sperm directly into the cytoplasm of an oocyte (egg cell) using a glass needle (injector), thereby increasing the likelihood of fertilization when there are abnormalities in the number, quality, or function of the sperm. An oocyte, as presented in Figure 4.27, comprises three parts: *zona pellucida* (cell membrane), *oolemma* (vitelline membrane), and cytoplasm (vitelline). Zona pellucida is a thick, transparent, elastic membrane that is usually difficult to pierce at a low speed. The oolemma, on the other hand, is a highly elastic membrane covering the cytoplasm, which contains the living substance and genetic information of the oocyte. In the ICSI procedure, the injector will first puncture the zona pellucida and then will advance to break the oolemma. Once the oolemma has been punctured, the sperm inside the needle can be released into the deep area of the cytoplasm.

The difficulties of this procedure arise from puncturing the zona pellucida and oolemma. With a conventional (manual) process, deformation and vibration of the oocyte

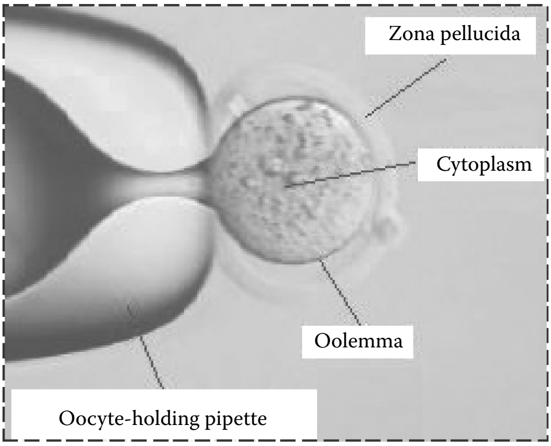


FIGURE 4.27
Structure of an oocyte.

may occur, increasing the internal pressure of the oocyte, inducing emission of cytoplasm upon penetration and eventually contributing to the oocyte death. To reduce the vibration and deformation of the oocyte, mercury has been used in ICSI experiments, but it can lead to serious health consequences, especially in the subsequent fetal development. Furthermore, the high elasticity of the oolemma does not allow the injector to readily penetrate the oocyte. The reported way of puncturing the oolemma is to apply negative pressure, thus drawing along a certain amount of cytoplasm, inducing further damage to the oocyte. Therefore, the conventional procedure depends heavily on the skill of the operator or embryologist.

The design of the ICSI installation to be presented shortly is intended to address the abovementioned issues of oocyte penetration to facilitate sperm injection. The requirement of this motion system is to execute a highly precise piercing motion through a soft, elastic, movable ball membrane of oocyte with a diameter of about 100 μm with a needle with a diameter of 10 μm without causing damage to the oocyte.

The actuation system to be applied is a linear reciprocating motion because this motion will penetrate the zona pellucida and oolemma in a gradual step-wise manner rather than a lump motion that is more destructive to the oocyte.

In this application, the resolution of the motion is on the order of a micrometer, and it is therefore reasonably safe to design the actuator to perform motion down to submicrometer resolution. To meet this requirement, a piezoelectric actuator is chosen as the actuator. To realize the linear reciprocating motion with high precision, the piezoelectric actuator is design in a stack construction. This construction has the advantage of generating larger displacement (due to its serial mechanical construction) with smaller electrical excitation (due to its parallel electrical connection). One end of the actuator is fixed, and the other end is in motion. The injector is attached to the motion end of the actuator.

The control of the ICSI installation is essentially a mechatronic control problem. The mathematical model of the actuator is determined such that it represents the performance of the actuator and facilitates the controller design. One has to make a compromise between a simple model for noncomplicated controller design and accurate modeling for a satisfactory result. Although the order of the actuator may be high, it is generally acceptable to assume the actuator as a second-order system. The nonlinearity of the actuator is then incorporated in one term of the model accordingly.

In this case, the following model will be used:

$$m\ddot{x} = -K_f\dot{x} - K_g x + K_e(u(t) - F) \tag{4.50}$$

where $u(t)$ is the time-varying motor terminal voltage, $x(t)$ is the position, K_f is the damping coefficient produced by the motor, K_g is the mechanical stiffness, K_e is the input control coefficient, m is the effective mass, and F is the system nonlinear disturbance.

The resolution of the sensor should be finer than the actuator; otherwise, the sensor is unable to sense the motion of the actuator. Because, in this case study, the motion of the actuator is in submicrometer resolution and because the position is the parameter of interest, a linear variable differential transducer (LVDT) is used as the sensor (Figure 4.28). As can be found with many sensor manufacturers, LVDT works at typically nanometer-level resolution. The LVDT is to be attached to the piezoelectric actuator at the motion end of the actuator. The sensing signal of the LVDT is first amplified to attain a readable level. To increase the sensitivity of the sensor, it can be incorporated into a Wheatstone bridge circuit.

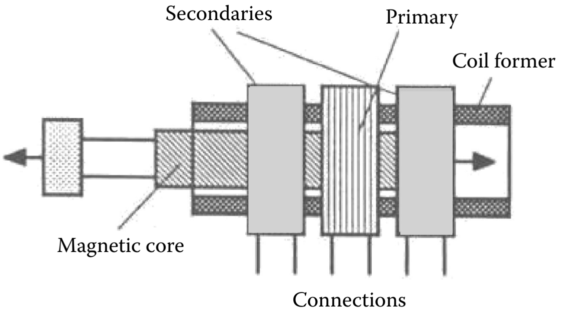


FIGURE 4.28
Linear variable differential transducer (LVDT).

The system can now be configured in a closed-loop manner. The piezoelectric stack actuator is the actuator in the system, and the LVDT is the sensor. The oocyte is the load, which, in this case study, is not considered to be part of the control system. It is assumed that there is no disturbance to the system. The controller is realized via a computer, on which the control algorithm is to be designed. The entire system is shown in Figure 4.29.

The actuator used in the installation is a stack piezoelectric actuator, which has a travel length of 80 μm, and it is equipped with LVDT sensor with an effective resolution of 5 nm. The dSPACE control development and rapid prototyping platform is used as the interface between the actuation and sensing components and the computer to allow a closed-loop control of the system. MATLAB®/Simulink® can be used from within a dSPACE environment. The entire software design for Simulink is shown in Figure 4.30. The position and switch signals are fed back to the control system by the analog input and digital input of the dSPACE card, respectively. The trajectory mode is forwarded to the adaptive controller as a reference signal, and then the controller output is sent out to the piezo actuator.

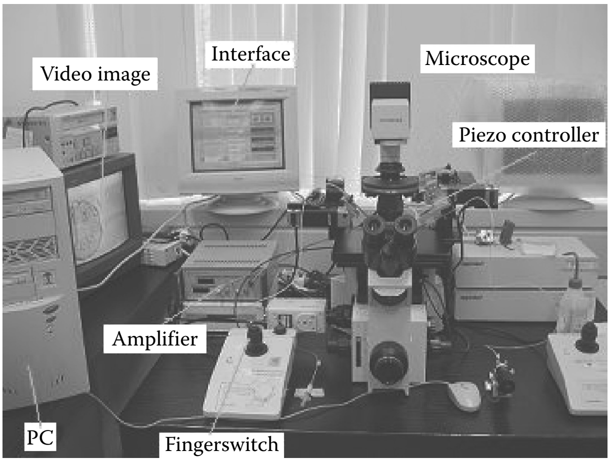


FIGURE 4.29
ICSI installation.

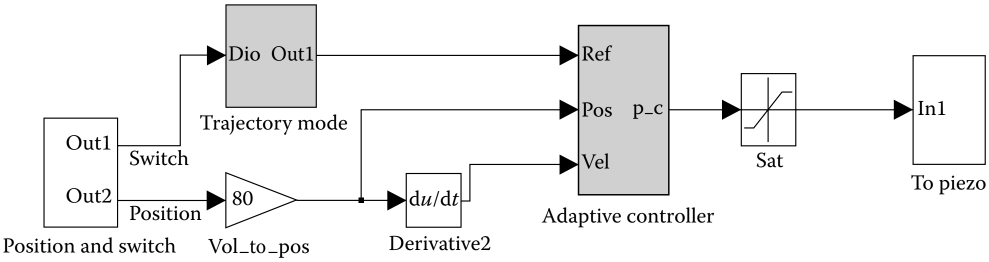


FIGURE 4.30
Software design for Simulink model.

The control system is implemented by running Real-Time Windows of MATLAB as presented in Figure 4.30.

A plausible reference signal for tracking for this application is a sinusoidal trajectory of $A \sin \omega t$, where A is 3 μm, and ω is 6 rad/sec.

Establishing the system model allows simulation of system behavior and of system response to control action. The nonlinearity of the system is represented in the following form:

$$F = (\sigma_1 + \sigma_2)\dot{x} + \sigma_0 z - \sigma_1 \frac{|\dot{x}|}{h(\dot{x})} z = (\sigma_1 + \sigma_2)\dot{x} + F_d(z, \dot{x}) \tag{4.51}$$

The first part, $(\sigma_1 + \sigma_2)\dot{x}$, is a simple function of the velocity, and the second part, $\left(\sigma_0 - \frac{\sigma_1 |\dot{x}|}{h(\dot{x})}\right) z$, is scaled by the z due to the dynamical perturbations in hysteresis.

To build the model of the actuator, the experimental data from the piezoelectric actuator is first collected. This procedure, known as system identification, includes sending an input signal to the actuator and recording the output signal from the actuator through the sensor. A chirp signal is selected because it allows using a limited amount of data with specified frequencies being put through the system. The sample rate is 2000 Hz, and the frequency of the chirp signal is varied from 0 to 200 Hz to suit the ICSI application. The dominant linear model is obtained to be

$$\ddot{x} = -1.08 \times 10^2 \dot{x} - 5.97 \times 10^5 x + 4.29 \times 10^6 u \tag{4.52}$$

The tracking control problem is stated as follows: *find a control mechanism for every bounded smooth output reference $x_d(t)$ with bounded time derivatives so that the controlled output $x(t)$ converges to the reference $x_d(t)$ as closely as possible.*

Controlling the system using PID control (as discussed in Section 4.3) entails tuning the gain of the PID control (K_P, K_I, and K_D) to achieve desired performance. With proper tuning using as discussed in Section 4.3, the tracking error can be minimized to about 1.0 μm as presented in Figure 4.31.

To improve the system even further, an adaptive controller can be designed based on the actuator model, in this case, to address the nonlinearity of the system.

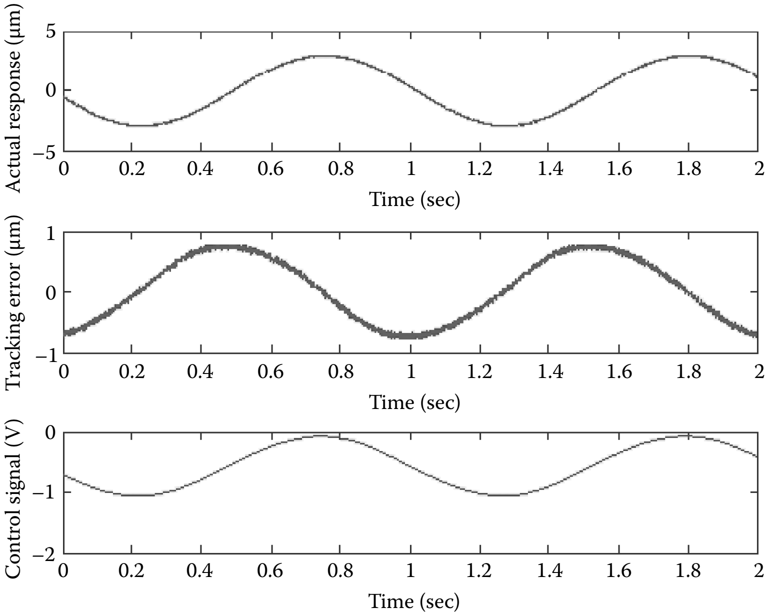


FIGURE 4.31
Sine wave response of the ICSI system when PID control is used as the controller.

The design is started by defining the position tracking error $e(t)$ and the corresponding filtered tracking error $s(t)$ as follows:

$$e(t) = x_d(t) - x(t) \tag{4.53}$$

$$s(t) = K_I \int_0^t e(\tau)\,d\tau + K_P e + \dot{e} \tag{4.54}$$

The dynamics of Equation 4.50 in terms of $s(t)$ is

$$\frac{m}{K_e}\dot{s} = \frac{m}{K_e}(K_I e + K_P \dot{e} + \ddot{x}_d) + \frac{K_f}{K_e}\dot{x} + \frac{K_g}{K_e}x - (u - F), \tag{4.55}$$

such that

$$\frac{m}{K_e}\dot{s} = \frac{m}{K_e}(K_I e + K_P \dot{e} + \ddot{x}_d) + \left(\frac{K_f}{K_e} + \sigma_1 + \sigma_2\right)\dot{x} + \frac{K_g}{K_e}x - u + F_d(z, \dot{x}). \tag{4.56}$$

By using straightforward exact model knowledge, the control input is

$$u = K_v s + a_m(K_I e + K_P \dot{e} + \ddot{x}_d) + a_{k\sigma}\dot{x} + a_{ge}x + F_d, \tag{4.57}$$

where $K_v > 0$ is a constant, and

$$a_m = \frac{m}{K_e} \tag{4.58}$$

$$a_{k\sigma} = \frac{K_f}{K_e} + \sigma_1 + \sigma_2 \tag{4.59}$$

$$a_{ge} = \frac{K_g}{K_e}. \tag{4.60}$$

Substituting the control input into the open-loop expression, the closed-loop filtered tracking error system is obtained as follows:

$$\frac{m}{K_e}\dot{s} = -K_v s. \tag{4.61}$$

Because $K_v > 0$, the resulting system is asymptotically stable. Unfortunately, the hysteresis is unknown a priori in practice. In addition, it is also difficult to obtain the precise values of m, K_e, K_f, and K_g. Motivated by this observation, the following replacements are put in place:

$$\frac{m}{K_e} \rightarrow \hat{a}_m \tag{4.62}$$

$$\frac{K_f}{K_e} + \sigma_1 + \sigma_2 \rightarrow \hat{a}_{k\sigma} \tag{4.63}$$

$$\frac{K_g}{K_e} \rightarrow \hat{a}_{ge} \tag{4.64}$$

$$F_d \rightarrow \hat{k}_1 sgn(s) + \hat{k}_2 \left|\dot{x}\right| sgn(r). \tag{4.65}$$

The adaptive control technique is then designed as follows:

$$u = K_v s + \hat{a}_m(K_I e + K_P \dot{e} + \ddot{x}_d) + \hat{a}_{k\sigma}\dot{x} + \hat{a}_{ge}x + \hat{k}_1 sgn(s) + \hat{k}_2 \left|\dot{x}\right| sgn(s). \tag{4.66}$$

The adaptive controller as presented in Equation 4.66 is applied to control the piezoelectric actuator. The parameters of the controller are selected in which $K_v = 10^{-5}$, $K_I = 4 \times 10^5$, and $K_P = 10^2$. The initial values based on the identified model of Equation 4.52 are $\hat{a}_m(0) = 2.33 \times 10^{-7}$, $\hat{a}_{k\sigma}(0) = 2.52 \times 10^{-4}$, and $\hat{a}_{ge}(0) = 1.18 \times 10^{-1}$; furthermore, $\hat{k}_1 = 10^{-7}$ and $\hat{k}_2 = 10^{-8}$. Due to the inherent unmodeled high-frequency dynamics in the mechanical

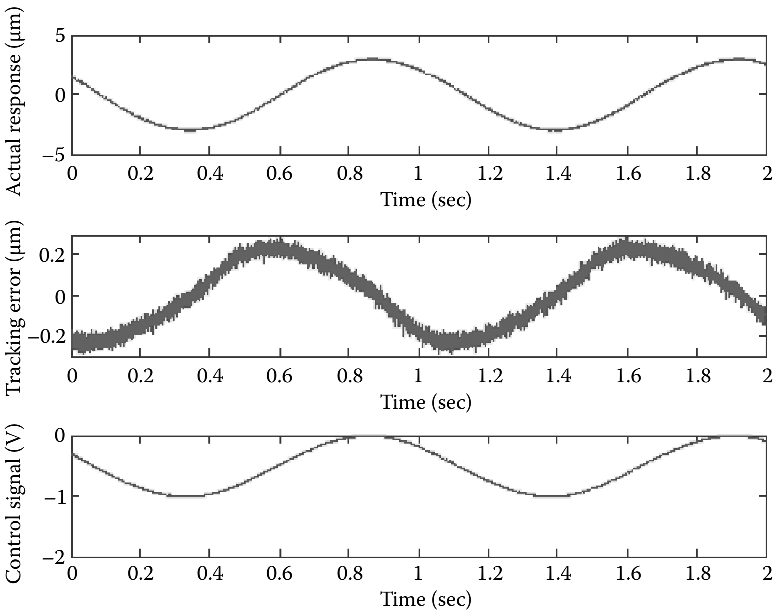


FIGURE 4.32
Sine wave response of the ICSI system in which adaptive control is used as the controller.

structure and other components in the system, which should not be excited, small adaptation factors are used as follows:

$$\gamma_1 = \gamma_2 = \gamma_3 = 10^{-22} \tag{4.67}$$

$$\gamma_4 = \gamma_5 = 10^{-20} \tag{4.68}$$

$$\gamma_{11} = \gamma_{21} = \gamma_{31} = \gamma_{41} = \gamma_{51} = 10^{-4} \tag{4.69}$$

Using the same reference signal of sinusoidal trajectory as used in the PID-controlled system above, that is, in which the signal is $A \sin \omega t$, A is 3 μm, and ω is 6 rad/sec, the adaptive controller is implemented to control the same ICSI system. Figure 4.32 shows the performance of the system with the adaptive control. The tracking error is about 0.3 μm, significantly better than using PID control.

Although the PID control can indeed achieve a rather satisfactory result with a rather simple method, a dedicated controller, such as adaptive control, to address specific needs of a particular mechatronic system is able to achieve a better performance at the expense of a more complicated control algorithm.

4.7 Conclusions

This chapter addressed control issues in mechatronic systems, including various examples and implementations of the control systems for illustration as well as standards and guides to common control software and algorithms. It provided a discussion on what mechatronic systems are and what made them gain widespread use in less than 40 years with some illustrative examples. A brief overview of control systems was presented with some techniques of control that were necessary for the subsequent discussions on the control of mechatronic systems. The focus of the discussions was the implementation issues, which included how to implement and the challenges in implementations. Case studies—from real implementations—at the end of the chapter provided a glimpse of control of mechatronic systems in practice.

Bibliography

American Society for Reproductive Medicine, "Patient's fact sheet: Intra-cytoplasmic sperm injection (ICSI)," retrieved from http://fwivf.com/new-patient/documents/ICSI.pdf, 2008.

ATSDR, *Toxicological Profile of Mercury*, Agency for Toxic Substances and Disease Registry, Atlanta, GA, 1999. Canudas-De-Wit C., H. Olsson, K. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.

Chen, F., H. Xie, and G. K. Fedder, "A MEMS-based monolithic electrostatic microactuator for ultralow magnetic disk head fly height control," *IEEE Transactions on Magnetics*, vol. 37, no. 4, pp. 1915–1918, 2001.

Choi, G. H., J. H. Oh, and G. S. Choi, "Repetitive tracking control of a coarse-fine actuator," *Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, GA, pp. 335–340, 1999.

de Silva, C. W., *Sensors and Actuators—Engineering System Instrumentation*, 2nd Edition, Taylor & Francis, CRC Press, Boca Raton, FL, 2015.

Ediz, K. and N. Olgac, "Effect of mercury column on the microdynamics of the piezo-driven pipettes," *Journal of Biomechanical Engineering*, vol. 127, pp. 531–535, 2005.

Foulard, C., S. Gentil, and J. P. Sandraz, *Commande et regulation par calculateur numeriques*, Eyrolles, Paris, 1977.

Goldfarb, M. and N. Celanovic, "Modeling piezoelectric stack actuators for control of micromanipulation," *IEEE Control Systems*, vol. 17, pp. 69–79, 1997.

Harashima, F., M. Tomizuka, and T. Fukuda, "Mechatronics—What is it, why, and how?," *IEEE Trans. Mechatrn.*, vol. 1, no. 1, pp. 1–4, 1996.

Lahmar, D., L. Loron, and M. Bonis, "Asservissement en tres basse vitesse d'un moteur autosynchrone," *Conf Canadienne sur l'Automatisation Industrielle*, Montreal, 1992.

Low, T. S. and W. Guo, "Modeling of a three-layer piezoelectric bimorph beam with hysteresis," *Journal of Microelectromechanical Systems*, vol. 4, no. 4, pp. 230–237, 1995.

Mekid, S. and O. Olejniczak, "High precision linear slide. Part II: Control and measurements," *International Journal of Machine Tool and Manufacture*, vol. 40, pp. 1051–1064, 2000.

Palermo, G., H. Joris, M. P. Derde, and A. C. van Steirteghem, "Pregnancies after intra-cytoplasmic injection of single spermatozoon into an oocyte," *Lancet*, vol. 340, pp. 17–18, 1992.

Robert, G., D. Damjanovic, and N. Setter, "Separation of nonlinear and friction-like contributions to the piezoelectric hysteresis," *12th IEEE International Symposium on Applications of Ferroelectrics*, vol. 2, pp. 699–702, Honolulu, USA, 2000.

Tan, K. K., H. F. Dou, Y. Q. Chen, and T. H. Lee, "High precision linear motor control via relay tuning and iterative learning based on zero-phase filtering," *IEEE Transactions on Control Systems Technology*, vol. 9. pp. 244–253, 2001.

Tan, K. K., S. N. Huang, T. H. Lee, S. J. Chin, and S. Y. Lim, "Adaptive robust motion control for precise trajectory tracking applications," *ISA Transactions*, vol. 40, pp. 17–29, 2001.

Tan, K. K., T. H. Lee, S. N. Huang, and X. Jiang, "Friction modeling and adaptive compensation using a relay feedback approach," *IEEE Transactions on Industrial Electronics*, vol. 48, pp. 169–176, 2001.

Tan, K. K., S. C. Ng, and Y. Xie, "Optimal intra-cytoplasmic sperm injection with a piezo micromanipulation," *Proceedings of the 4th World Congress on Intelligent Control and Automation*, pp. 1120–1123, 2002.

Wulp, H., *Piezo-Driven Stages for Nanopositioning with Extreme Stability: Theoretical Aspects and Practical Design Considerations*, Delft University of Technology, Delft, 1997.

Yanigada, K., H. Katayose, H. Yazawa, Y. Kimura, K. Konnai, and A. Sato, "The usefulness of a piezo-micromanipulator in intra-cytoplasmic sperm injection in humans," *Human Reproduction*, vol. 14, no. 2, pp. 448–453, 1998.

5

Introduction to Sensors and Signal Processing

Diogo Montalvão

CONTENTS

5.1 Introduction ... 127
5.2 Signals ... 127
  5.2.1 Types of Time Signals and Waveforms ... 127
  5.2.2 Harmonic Signals ... 129
    5.2.2.1 Definition ... 129
    5.2.2.2 Harmonic Motion in the Argand Plane ... 130
    5.2.2.3 Differentiation of Harmonic Signals ... 132
  5.2.3 Quantification of Energy in a Signal: RMS ... 134
  5.2.4 Useful Relationships and Common Waveforms ... 138
5.3 Fourier Analysis ... 138
  5.3.1 Introduction ... 138
  5.3.2 Fourier Transform ... 141
  5.3.3 Fourier Transform Application Example ... 143
  5.3.4 Basics of the Discrete and Fast Fourier Transforms ... 147
    5.3.4.1 Sampling Frequency ... 147
    5.3.4.2 Discrete Fourier Transform ... 148
    5.3.4.3 Fast Fourier Transform ... 149
5.4 Signal Processing ... 150
  5.4.1 Aliasing ... 150
  5.4.2 Quantization Errors ... 155
  5.4.3 Leakage and Windowing ... 155
  5.4.4 Convolution ... 160
  5.4.5 Random Signals ... 161
    5.4.5.1 Auto-Spectrum, Power Spectrum, and Cross-Spectrum ... 161
    5.4.5.2 Estimators ... 165
    5.4.5.3 Ensemble Averaging ... 167
  5.4.6 Butterworth Filter ... 168
  5.4.7 Smoothing Filters ... 173
    5.4.7.1 Moving Average ... 174
    5.4.7.2 Savitzky–Golay ... 175
5.5 Sensors ... 177
  5.5.1 Accelerometers ... 177
    5.5.1.1 Piezoelectric Accelerometers ... 177
    5.5.1.2 Piezoresistive and Capacitive Accelerometers ... 180
  5.5.2 Velocity Transducers ... 181
    5.5.2.1 Laser Doppler Velocimeters ... 181
    5.5.2.2 Tachometers ... 183

- 5.5.3 Displacement Transducers 184
  - 5.5.3.1 LVDT 184
  - 5.5.3.2 Laser 186
  - 5.5.3.3 Proximity Probes 187
  - 5.5.3.4 MEMS Sensors 188
- 5.5.4 Strain Gauges 189
- 5.5.5 Load Cells 193
  - 5.5.5.1 Piezoelectric Force Transducers 193
  - 5.5.5.2 Strain Gauge–Based Load Cells 195
  - 5.5.5.3 Calibration of a Pair Force Transducer versus Accelerometer 197
- 5.5.6 Temperature Sensors 198
  - 5.5.6.1 Thermocouple 199
  - 5.5.6.2 Thermistors and Resistance Thermometers 199
  - 5.5.6.3 Bimetallic Thermometers 200
  - 5.5.6.4 Infrared Sensors 200
- 5.5.7 Flow Sensors 201
  - 5.5.7.1 Venturi Tube 202
  - 5.5.7.2 Pitot Tube 202
  - 5.5.7.3 Anemometers and Angular Momentum Flow Meters 203
  - 5.5.7.4 Rotameter 205
  - 5.5.7.5 Other Flow Measurement Sensors 205
- 5.5.8 Pressure Transducers 206
- 5.5.9 Ultrasonic Sensors 207
  - 5.5.9.1 Encoders 208
  - 5.5.9.2 Incremental Encoders 208
  - 5.5.9.3 Absolute Encoders 209
- 5.5.10 Other Sensors 210

5.6 Logarithmic Scales 211
- 5.6.1 Decibel 211
  - 5.6.1.1 Power Quantities 211
  - 5.6.1.2 Root-Power Quantities 213
  - 5.6.1.3 Linear versus Logarithmic Frequency Plots 214
  - 5.6.1.4 dB Reference Values 214
  - 5.6.1.5 Comparison between the Power and Root-Power dB Scales 214
- 5.6.2 Octave 215

5.7 Conclusions 218

References 218

SUMMARY This chapter presents the fundamentals of sensors and signal processing with an emphasis on mechatronics and applications. Sensors are used to measure signals that present changes in the time domain, for example, waveforms or digital steps. Different technologies have been developed over the years in order to sense many different physical quantities, such as temperature, flow, force, acceleration, position, sound pressure, and intensity of light, among others. Because of their varying nature, all these quantities may be measured under the form of waveforms. However, waveforms—which are analog signals—are often difficult to interpret in the time domain, and a transformation into the frequency domain is required. The Fourier transform still is the most popular technique used today for converting a time signal into a frequency spectrum. Nevertheless, in

signal processing, an analog-to-digital conversion (ADC) of the time signal is required at some stage even if the Fourier transform is not used. When proper treatment and filtering approaches are not followed, important features in the signal may be attenuated, and others may be falsely indicated. This chapter discusses how signals can be measured in order to avoid common pitfalls in signal acquisition and processing. The theoretical background is set in a comprehensive yet practical way.

5.1 Introduction

Sensors exist in every system that interacts with the surrounding world. Any nonarbitrary decision requires a sensor to collect, transmit, and process data somewhere. We have all been using sensors since we were born: From our eyes to our nervous system, our body is a complex network of sensors that continuously monitor and send signals to our brain for processing and analysis. The human body has sensors that are capable of acquiring color images, sound, flavor, odor, texture, and temperature. However, it is not equipped with a sensor that is capable of detecting a magnetic field like a compass does. On the other hand, the human body's sensors have some known limitations as well. In acoustics, a rule of thumb is the rule of 20: Roughly speaking, a healthy young human can detect sounds within the 20 Hz to 20 kHz frequency range for pressure waves between 20 μPa and 20 Pa (threshold of pain). Still, a dog can hear sounds up to 50 kHz, and a bat is capable of detecting sounds up to 100 kHz [1,2].

Likewise, applications in mechatronics require sensors to capture light, sound, motion, temperature, force, flow, current, etc. With the development of powerful energy and memory storage units, wireless technology, and fast microcomputers, data can be continuously monitored, processed, stored, and made available for immediate use without the need of further human interaction. However, raw signals are not necessarily useful before being translated into something "readable." Signal processing is the discipline that allows transforming a signal into something useful without losing relevant information.

This chapter covers the fundamental aspects of signal processing and sensor technology, including an introduction to the mathematical representation of harmonic and complex time signals, Fourier analysis and its implementation, an introduction to random signals, sampling and aliasing, windowing and leakage, filtering and smoothing, typical sensors and applications, and representation of signals using logarithmic scales.

5.2 Signals

5.2.1 Types of Time Signals and Waveforms

A signal can be defined, in abstract, as "a function that conveys information about the behavior or attributes of some phenomenon" [3]. Generally, the nature of a signal is electromagnetic (e.g., the change in current or voltage in an AC signal), mechanical (e.g., the change in velocity of an oscillating pendulum), acoustical (e.g., the change in sound pressure during speech), or visual (e.g., the change in light intensity when looking at a star in the sky), among others.

Time signals all share one thing in common regardless of their nature: They all describe the variation of the amplitude of a quantity with time. The problem, however, is how to extract information from these functions in an intelligible and useful way. Signals may be grouped into *deterministic* and *nondeterministic* categories. Deterministic signals may be represented by analytical equations. Hence, it is possible to predict, with exactitude, how a signal will be in a given moment in the future as long as no changes are introduced into the system. On the other hand, nondeterministic signals cannot be represented by analytical equations. However, a system that produces nondeterministic signals can usually be treated as stochastic. Forecasts can be done based on probability and statistical methods, but it is not possible to predict, with exactitude, how the signal will be in a given moment in the future. In other words, if the change in the time signal is perfectly known, even if the signal is very irregular, then the signal is said to be *deterministic*. If the change in the time signal is not perfectly known but if it can be considered stochastic, that is, the subsequent state of the system is determined probabilistically, then the signal is said to be *nondeterministic*.

Nondeterministic signals are also called *random* and can be divided into *stationary* and *nonstationary* signals. A stationary signal is one that is a stochastic process whose joint probability distribution does not change when shifted in time. Consequently, parameters such as the mean and variance, if they are present, also do not change over time and do not follow any trends [4]. One example of a nondeterministic stationary signal is the sound of rain hitting the roof of a car.

Deterministic signals are often divided into two main categories: *periodic* and *nonperiodic*. The first ones are those that present the same amplitude, direction, and position after a complete cycle in time is completed (*period*). The simplest form of a deterministic stationary signal is the *simple harmonic*, which is characterized by amplitude, frequency, and phase (see Section 5.2.2). Examples of predominantly harmonic signals are those generated

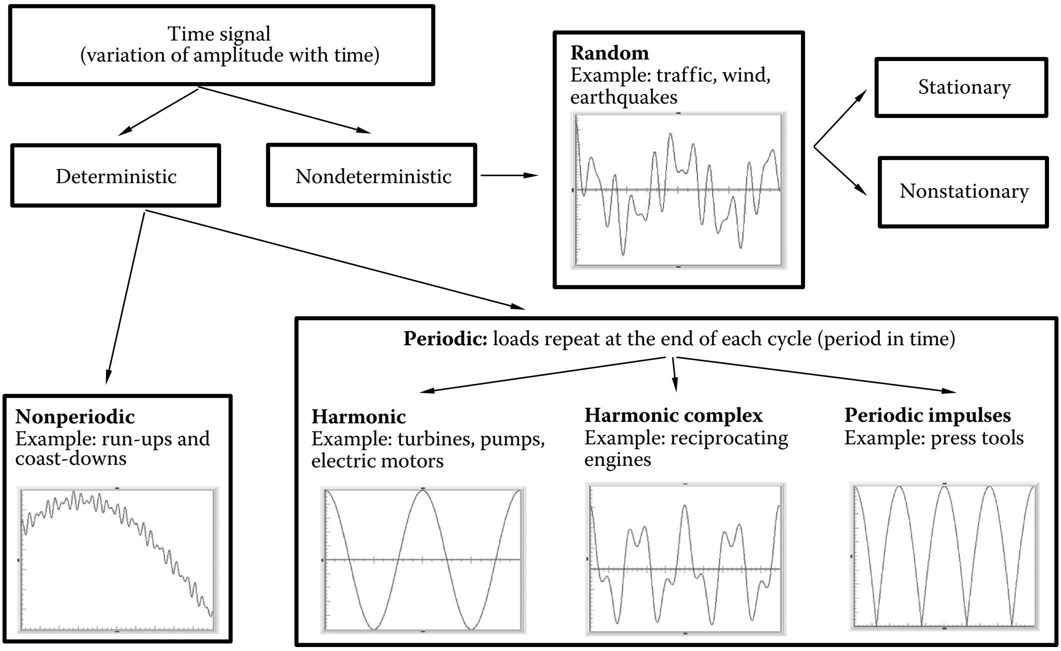

Time signal
(variation of amplitude with time)
Deterministic
Nondeterministic
Random
Example: traffic, wind, earthquakes
Stationary
Nonstationary
Periodic: loads repeat at the end of each cycle (period in time)
Nonperiodic
Example: run-ups and coast-downs
Harmonic
Example: turbines, pumps, electric motors
Harmonic complex
Example: reciprocating engines
Periodic impulses
Example: press tools

FIGURE 5.1
Types of time signals.

from, for example, high-speed turbines, pumps, electric motors, AC electric currents, or the sound produced from a human while whistling. Other systems, such as reciprocating engines (as in gasoline- and diesel-powered cars), still produce deterministic signals but, because many different parts are rotating at different speeds, they produce simultaneous harmonic excitations at various frequencies. Nevertheless, these *complex harmonic* signals can be decomposed and represented by a sum of simple harmonic signals using, for example, a Fourier analysis (see Section 5.3).

Nonperiodic signals are *transient* signals that can still be described by a deterministic approach (for example, some random nonstationary signals can be composed of a series of transient signals, yet they are not deterministic because of their random nature). These can be generated during the run-up and coast-down of machines or from periodic impulses, for example, from press tools or musical percussion systems. The Dirac is a special case of a nonperiodic impulse that still produces a deterministic signal once the impulse response function (IRF) is known (see Section 5.4.4), for instance, transient bursts observed in some hi-fi systems during turn-on and turn-off.

Figure 5.1 is a block diagram illustrating the previous discussion on how signals are grouped.

5.2.2 Harmonic Signals

5.2.2.1 Definition

Simple harmonic (often simply referred to as *harmonic*) signals are used to describe many deterministic systems even if the functions are, from the point of view of the analyzer, apparently rather complex (e.g., the case of complex harmonic waveforms). Harmonic signals are trigonometric circular functions that can be represented either as a function of time (amplitude versus time) or as rotating vectors in the Argand plane (real part versus imaginary part). This latter representation is going to be discussed in more detail later on.

Before introducing the mathematical equations describing the phenomenon, harmonic motion can be described by means of a simple example. First, let us imagine the simple case in which we have a mass m being hung with a spring with stiffness k as represented in Figure 5.2a. If the mass is disturbed from equilibrium and then suddenly released as shown in Figure 5.2b, it will start oscillating up and down about its equilibrium position as illustrated in Figure 5.2c. The distance X from the equilibrium position to one of the

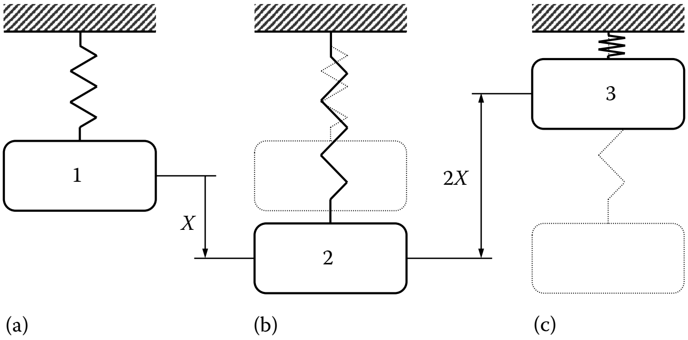
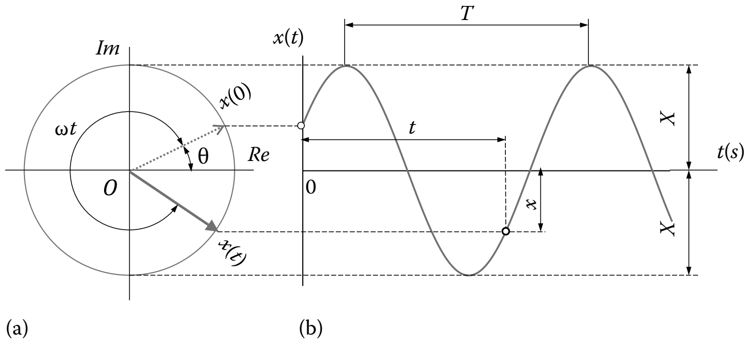


FIGURE 5.2
Oscillatory motion in a single degree of freedom (SDOF) mass–spring system. (a) Equilibrium, (b) disturbance, and (c) oscillation.



(a) (b)

FIGURE 5.3
Simple harmonic signal represented in the Argand plane (a) and in an amplitude versus time plot (b).

extremes is the *peak amplitude* or *magnitude*, and the total distance $2X$ traveled from one extreme to the other is the *peak-to-peak amplitude*.

Thereby, the harmonic motion of this oscillating mass–spring system can be described as a sinusoidal function that depends on time t:

$$x(t) = X\sin(\omega t + \theta) \tag{5.1}$$

where X is the amplitude, ω is the angular frequency in rad/s, and θ is the phase angle in rad. This is represented in Figure 5.3b, where T is the period (time required for a complete cycle of oscillation to repeat itself).

The period T is related to the angular frequency by

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \tag{5.2}$$

where f is frequency in cycles per second or Hertz (Hz). The frequency is more often expressed in Hz rather than in rad/s because it is easier to understand although the rad/s is, in fact, the one used in the International System (SI) of units.

Figure 5.3a represents the same signal as Figure 5.3b but, in this case, in the Argand plane. Instead of the signal being represented by the trace of a point that moves up and down as time goes by, a vector is used. Its size is the amplitude X. This vector rotates about the origin O of a Cartesian system of coordinates with real (horizontal) and imaginary (vertical) axes with constant angular velocity ω. The tip of the vector moves over a circle with radius X, and the angle it makes with the horizontal axis at $t = 0\text{s}$ is defined as the phase angle θ.

The amplitude of the harmonic signal shown in Figure 5.3. X is also called *peak amplitude* x_{PK}, and the value $2X$ is called the *amplitude peak-to-peak* x_{PK-PK}. This is equally represented in Figure 5.2.

5.2.2.2 Harmonic Motion in the Argand Plane

The simple harmonic motion, as written by Equation 5.1, is a trigonometric function that can be alternatively represented as a circular vector rotating in the Argand plane. It will be shown that this is a much more convenient form of representation because the arithmetic and differentiation of trigonometric functions is much more difficult in comparison to complex functions.

The Argand plane is composed of two mutually perpendicular orthogonal axes that we call real (*x*-axis) and imaginary (*y*-axis). A period of oscillation is completed when a vector completes a full rotation about the origin *O* of the Argand plane system of coordinates (dark vector in Figure 5.4).

Now, we must recall that a vector is a quantity that is defined by magnitude and direction (which includes line of action and sense) and can be decomposed into coordinates. Two well-known systems of coordinates are the Cartesian and Polar. In the Cartesian system of coordinates, a vector $x(t)$ is decomposed into its projections in the horizontal and vertical axes, which, in the Argand Plane, are $x_{Re}(t)$ and $x_{Im}(t)$, respectively:

$$x_{Re}(t) = X\cos(\omega t + \theta) \tag{5.3}$$

and

$$x_{Im}(t) = X\sin(\omega t + \theta) \tag{5.4}$$

One interesting aspect is that Equations 5.1 and 5.4 are exactly the same. Also, Equation 5.4 is equivalent to Equation 5.3 with a 90° (π/2 rad) phase shift. These vectors are represented in light grey in Figure 5.4.

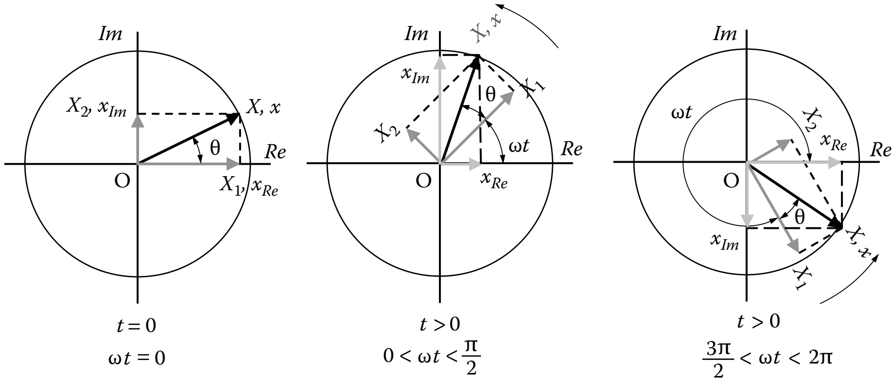
The polar coordinates are, essentially, the magnitude X and the phase angle θ. These can be translated into the Cartesian system of coordinates previously defined by

$$X = \sqrt{x_{Re}^2(t) + x_{Im}^2(t)} = \sqrt{X_1^2 + X_2^2} \tag{5.5}$$

and

$$\theta = \tan^{-1}\frac{x_{Im}(t)}{x_{Re}(t)} = \tan^{-1}\frac{X_2}{X_1} \tag{5.6}$$

Quantities X_1 and X_2 are, respectively, the real and imaginary components of x at $t = 0$s. Although $x_{Re}(t)$ and $x_{Im}(t)$ are functions that depend on time (thus, quantities that change with time), X_1 and X_2 are constant quantities.



$t = 0$, $\omega t = 0$ | $t > 0$, $0 < \omega t < \frac{\pi}{2}$ | $t > 0$, $\frac{3\pi}{2} < \omega t < 2\pi$

FIGURE 5.4
Representation of a circular vector rotating in the Argand plane.

The quantity X expressed in Equation 5.5 is a real number: It is a scalar. Thus, it only represents the magnitude of $x(t)$ but has no information about the direction. When we incorporate the direction (which is changing with time because the dark vector in Figure 5.4 is rotating about O), X becomes $x(t)$:

$$x(t) = X_{Re}(t) + iX_{Im}(t) \tag{5.7}$$

where $i = \sqrt{-1}$ is the imaginary number. When X is written as a complex number, it is, in fact, a vector in the Argand plane. Its real part represents its projection on the horizontal axis of the Argand plane, and its imaginary part represents its projection on the vertical axis. If we now replace the quantities in Equation 5.7 with Equations 5.3 and 5.4, we obtain:

$$x(t) = X\cos(\omega t + \theta) + iX\sin(\omega t + \theta) = X\text{cis}(\omega t + \theta) \tag{5.8}$$

which is known to be equivalent to

$$x(t) = Xe^{i(\omega t + \theta)} \tag{5.9}$$

In this latter formulation, $x(t)$ is, in fact, represented in terms of polar components: the amplitude (or magnitude) X, which is constant, and the angle formed by the vector with the real horizontal axis, $\omega t + \theta$, which changes with time.

5.2.2.3 Differentiation of Harmonic Signals

One interesting property about harmonic motion is that it can be easily differentiated. This can be useful, for instance, when comparing signals obtained from different sensors. For example, when measuring structural vibrations, the most common sensor used is the accelerometer (see Section 5.5.1). This sensor measures the acceleration of oscillating mechanical systems. However, vibration level limits are many times specified as velocities. So, how can acceleration be translated into velocity? In harmonic motion, this is quite straightforward as will be shown.

First, let us recall that velocity $\dot{x}$ is the rate of change of position x with time, and acceleration $\ddot{x}$ is the rate of change of velocity $\dot{x}$ with time, that is,

$$\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t} \tag{5.10}$$

$$\ddot{x} = \frac{\mathrm{d}\dot{x}}{\mathrm{d}t} = \frac{\mathrm{d}^2x}{\mathrm{d}t^2} \tag{5.11}$$

Thus, if Equation 5.1 is the displacement, then from Equation 5.10 the velocity is

$$\dot{x}(t) = \omega X\cos(\omega t + \theta) = \omega X\sin\left[\omega t + \left(\theta + \frac{\pi}{2}\right)\right] \tag{5.12}$$

and from Equation 5.11 the acceleration is

$$\ddot{x}(t) = -\omega^2 X\sin(\omega t + \theta) = \omega^2 X\sin[\omega t + (\theta + \pi)] \tag{5.13}$$

Two important results can be extracted from this. The first one is that the amplitudes of displacement, velocity, and acceleration are related to one another by the angular frequency ω. The second one is that there is a phase angle of 90° (*quadrature*) between displacement and velocity and between velocity and acceleration, and there is a phase angle of 180° (*phase opposition*) between displacement and acceleration.

The amplitudes of velocity and acceleration are, respectively,

$$V = \omega X \qquad A = \omega V = \omega^2 X \tag{5.14}$$

and the phase angles of velocity and acceleration are, respectively,

$$\begin{aligned} \vartheta &= \theta + \frac{\pi}{2} \\ \alpha &= \vartheta + \frac{\pi}{2} = \theta + \pi \end{aligned} \tag{5.15}$$

Thus, harmonic signals can be differentiated (or integrated) by simply multiplying their amplitudes (or dividing them) by the angular frequency and adding (or subtracting) $\pi/2$ rad to the phase. This simple process is represented schematically in Figures 5.5 and 5.6.

The use of Equation 5.1 is not very practical, however, because the conversion involves two operations: conversion of amplitude and conversion of phase. The use of the complex notation (Equation 5.9) allows converting both the amplitude and phase in a single operation.

$$\begin{tikzcd} \boldsymbol{x} \arrow[r, shift left, "\frac{\mathrm{d}}{\mathrm{d}t}"] & \ddot{\boldsymbol{x}} \arrow[l, shift left, "\int \mathrm{d}t"] \arrow[r, shift left, "\frac{\mathrm{d}}{\mathrm{d}t}"] & \ddot{\boldsymbol{x}} \arrow[l, shift left, "\int \mathrm{d}t"] \end{tikzcd}$$

$$\begin{tikzcd} X \arrow[r, shift left, "\times\omega"] & V \arrow[l, shift left, "\div\omega"] \arrow[r, shift left, "\times\omega"] & A \arrow[l, shift left, "\div\omega"] \end{tikzcd}$$

$$\begin{tikzcd} \boldsymbol{\theta} \arrow[r, shift left, "+\frac{\pi}{2}"] & \boldsymbol{\vartheta} \arrow[l, shift left, "-\frac{\pi}{2}"] \arrow[r, shift left, "+\frac{\pi}{2}"] & \boldsymbol{\alpha} \arrow[l, shift left, "-\frac{\pi}{2}"] \end{tikzcd}$$

FIGURE 5.5
Relationships between displacement, velocity, and acceleration in a harmonic signal.

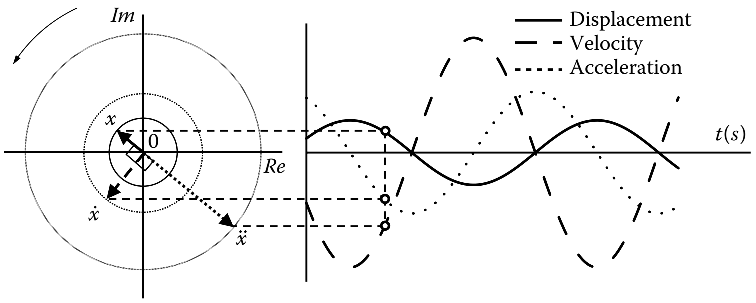


FIGURE 5.6
Graphical representation of the relationships between displacement, velocity, and acceleration in a harmonic signal.

$$\begin{tikzcd} x \arrow[r, shift left, "\frac{\mathrm{d}}{\mathrm{d}t}"] & \dot{x} \arrow[l, shift left, "\int \mathrm{d}t"] \arrow[r, shift left, "\frac{\mathrm{d}}{\mathrm{d}t}"] & \ddot{x} \arrow[l, shift left, "\int \mathrm{d}t"] \end{tikzcd}$$

$$\begin{tikzcd} x \arrow[r, shift left, "\times i\omega"] & \dot{x} \arrow[l, shift left, "\div i\omega"] \arrow[r, shift left, "\times i\omega"] & \ddot{x} \arrow[l, shift left, "\div i\omega"] \end{tikzcd}$$

FIGURE 5.7
Complex relationships between displacement, velocity, and acceleration in a harmonic signal.

If we now differentiate Equation 5.9, we obtain the complex forms for the velocity and acceleration:

$$\dot{x}(t) = i\omega X e^{i(\omega t+\theta)} \tag{5.16}$$

$$\ddot{x}(t) = (i\omega)^2 X e^{i(\omega t+\theta)} = -\omega^2 X e^{i(\omega t+\theta)} \tag{5.17}$$

The quantity $Xe^{i(\omega t+\theta)}$ in Equations 5.16 and 5.17 is the displacement $x(t)$ itself as given by Equation 5.9, thus

$$\dot{x}(t) = i\omega x(t) \tag{5.18}$$

$$\ddot{x}(t) = i\omega\dot{x}(t) = -\omega^2 x(t) \tag{5.19}$$

Thereby, complex notation requires only one operation in the differentiation of a harmonic signal: to multiply or to divide by $i\omega$. This is illustrated in Figure 5.7.

In conclusion, one advantage for using complex representation of harmonic signals is that a complex number, by being composed of real and imaginary parts, contains both information on magnitude and phase.

5.2.3 Quantification of Energy in a Signal: RMS

One of the problems associated with continuous time signals is related to the quantification of energy within the waveform—whether it is electromagnetic or mechanical. For example, one may need to know the power dissipated by the resistance in an electric circuit or if the global level of vibration of an engine is above an acceptable limit.

Let us consider two signals, signal A and signal B, as shown in Figure 5.8. One way of quantifying these signals would be by measuring their highest peaks: signal A's peak is 19.6, which is larger than signal B's peak of 16.4. However, signal B is larger than signal A most of the time. Hence, the quantification of a continuous time signal using just the peak as a measure is not adequate to show how much "energy" the signal contains. A better way of doing it is by using the root mean square (RMS). In this example, signal B's RMS value is 7.8, which is larger than signal A's RMS value of 6.9, contrary to what happens when the peaks are compared.

Before defining what the RMS value is, it is important to understand first why this is the preferred method to quantify, globally, a signal. Let us start by defining a harmonic time signal with unitary amplitude and plot the first two periods (Figure 5.9). The plot on the left (a) is a continuous waveform, representing an analog signal, and the plot on the right (b) is a discrete waveform, representing a digital signal.

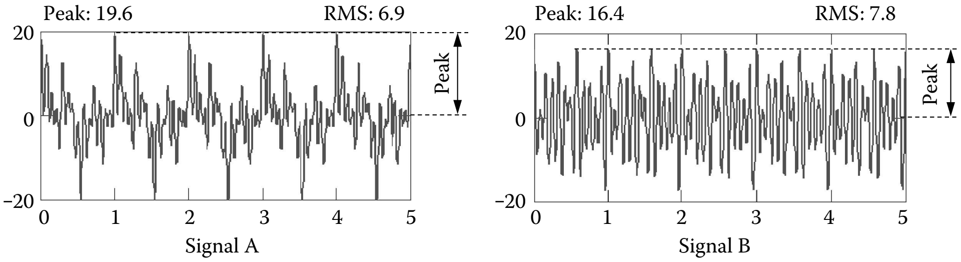


FIGURE 5.8
Continuous time signals with different peak and RMS values.

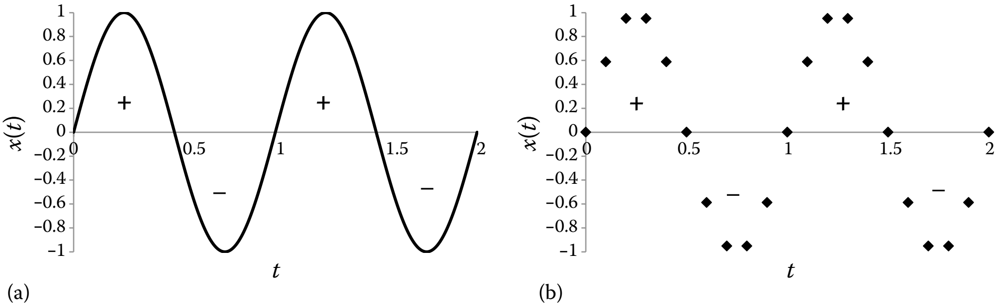


FIGURE 5.9
Plot of two periods of a harmonic time signal with amplitude 1 and frequency 1 Hz: (a) continuous (or analog) waveform and (b) discrete (or digital) waveform.

One way to quantify this signal would be to determine the average. By definition, the average (or arithmetic mean) is

$$\bar{x} = \frac{1}{T}\int_0^T x(t)\,\mathrm{d}t \tag{5.20}$$

for continuous functions (analog signal) and

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{5.21}$$

for discrete functions (digital signal), where T is the time span, and N is the number of points (samples) in the discrete waveform.

However, the average is not useful. Figure 5.9 also shows that this harmonic signal oscillates about zero with positive and negative values. Thus, when computing the average, it should be obvious that the positive values cancel the negative values. As a consequence, the average is zero for the harmonic time signal shown when an integer number of periods is measured (in this case, two) regardless of the amplitude, frequency, and phase.

The immediate suggestion to correct this problem is to determine the average of the absolute values. In this case, the original signal shown in Figure 5.9 takes the form shown in Figure 5.10.

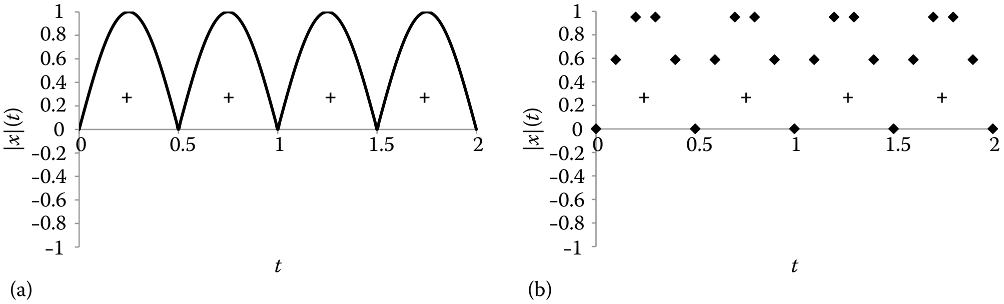


FIGURE 5.10
Plot of two periods of the average of the harmonic time signal represented in Figure 5.9: (a) continuous (or analog) waveform and (b) discrete (or digital) waveform.

The average of the absolute values is determined from

$$\bar{x}_{\text{abs}} = \frac{1}{T}\int_0^T |x(t)|\,\mathrm{d}t \tag{5.22}$$

for continuous functions (analog signal) and

$$\bar{x}_{\text{abs}} = \frac{1}{N}\sum_{i=1}^{N} |x_i| \tag{5.23}$$

for discrete functions (digital signal).

It is easy to show that the relationship between the peak amplitude and the average of the absolute values for a harmonic sine wave is

$$\bar{x}_{\text{abs}} = \frac{2}{\pi} x_{\text{PK}} \cong 0.637 x_{\text{PK}} \tag{5.24}$$

where x_{PK} is the peak amplitude. It should be noted that Equation 5.24 is not valid for complex waveforms and can only be applied to harmonic sine waves. If the average of the absolute values of a complex time signal is being determined, either Equation 5.22 or Equation 5.23 should be used in place of Equation 5.24.

The average of the absolute values is a first approach to quantify, globally, a time signal. However, it is clear from Figure 5.10 that the derivative is undefined at zero: The function is not continuous; thus it is not differentiable, which may bring some algebra problems.

An alternative consists of determining the RMS. The function is raised to the power of two so that all values become positive; it is integrated, and then the square root is determined. In this case, the original signal shown in Figure 5.9 takes the form shown in Figure 5.11.

The RMS is determined from

$$x_{\text{RMS}} = \sqrt{\frac{1}{T}\int_0^T x^2(t)\,\mathrm{d}t} \tag{5.25}$$

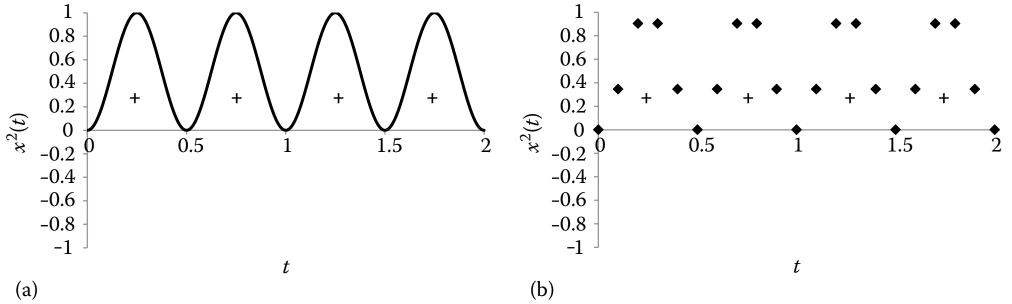


FIGURE 5.11
Plot of two periods of the harmonic time signal represented in Figure 5.9 raised to the power of two: (a) continuous (or analog) waveform and (b) discrete (or digital) waveform.

for continuous functions (analog signal) and

$$x_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{5.26}$$

for discrete functions (digital signal).

It can be shown that the relationship between the peak amplitude and RMS value for a harmonic sine wave is

$$x_{\mathrm{RMS}} = \frac{\sqrt{2}}{2} x_{\mathrm{PK}} \cong 0.707 x_{\mathrm{PK}} \tag{5.27}$$

It should be noted that Equation 5.27 is not valid for complex waveforms and can only be used with harmonic sine waves. If the RMS of a nonharmonic time signal is being determined, either Equation 5.25 or Equation 5.26 should be used in place of Equation 5.27.

Contrary to Figure 5.10 in which the derivative is undefined at zero, Figure 5.11 shows a continuous and differentiable function. This is one important result that encourages the use of the RMS over the use of the average of the absolute values.

Another interesting property is that the RMS is a statistical measure of the magnitude of a varying quantity. The standard deviation of variable x is defined as

$$\sigma_x = \sqrt{\frac{1}{T}\int_0^T (x(t) - \bar{x})^2\, \mathrm{d}t} \tag{5.28}$$

for continuous functions (analog signal) and

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})^2\, \mathrm{d}t} \tag{5.29}$$

for discrete functions (digital signal).

Consequently, for a harmonic signal with zero mean ($\bar{x} = 0$), Equations 5.28 and 5.29 turn into Equations 5.25 and 5.26, respectively, and the RMS value is the same as the standard deviation. On the other hand, if the average is not zero, the RMS is related to the standard deviation by the following equation [5]:

$$x_{\mathrm{RMS}}^2 = \bar{x}^2 + \sigma_x^2 \tag{5.30}$$

It is clear that the RMS value of a varying quantity is always greater than the average, in that the RMS includes the standard deviation as well.

In summary, the use of the RMS value to quantify a time signal has the following advantages:

1. Contrary to the use of the average, it can give a measure of the "amount of energy" in a signal.* For example, the power dissipated on an electrical resistor depends on the square of the current, and the energy stored in a spring depends on the square of the deformation. The use of the square root brings back the units we are used to instead of leaving them raised to the power of two.
2. The power of two of a harmonic time signal still is a continuous and limited function, which is differentiable, contrary to the average of the absolute values, which is not.
3. The RMS is a statistical measure of the magnitude of a varying quantity. For the particular case of a harmonic signal with zero average, it actually is the standard deviation.

5.2.4 Useful Relationships and Common Waveforms

Common waveforms, their shapes, equations, and RMS values are shown in Table 5.1. Except for the DC and pulse train waveforms, all other waveforms have zero mean, peak amplitude X, and peak-to-peak amplitude $2X$.

For harmonic signals, it is also useful that the relationships between RMS, peak, and peak-to-peak are known. For a sine wave, this is illustrated in Figure 5.12 (these relationships are not valid for nonharmonic signals).

5.3 Fourier Analysis

5.3.1 Introduction

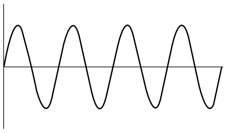
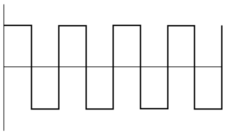
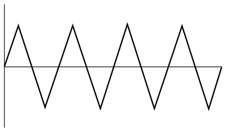
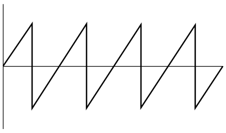
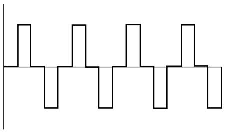
The Fourier transform is a well known and widely used mathematical tool nowadays, described in many textbooks (e.g., Ref. [6]). It is not the aim of this section to discuss the many different versions that have been developed over the years but solely to introduce it and to address a few of the most common problems associated with its practical use.

---

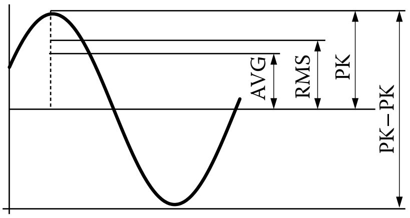
* It is now important to mention that the RMS value is not, in rigor, the same as energy. Energy is measured in J (Joules), and the RMS value has the same units as those in the waveform, for example, V (Volts) or µm (micrometers).

TABLE 5.1

Common Waveforms and Their Shapes

Waveform	Shape	Equation	RMS
Sine wave		$x(t) = X\sin(\omega t + \theta)$	$x_{\text{RMS}} = \frac{\sqrt{2}}{2}X$
Square wave		$x(t) = \begin{cases} X, & \langle ft \rangle < 0.5 \\ -X, & \langle ft \rangle > 0.5 \end{cases}$	$x_{\text{RMS}} = X$
Triangular wave		$x(t) = 2\lvert 2X\langle ft + s \rangle - X \rvert - X$ (s = 75% is the phase shift for the shape shown)	$x_{\text{RMS}} = \frac{\sqrt{3}}{3}X$
Sawtooth wave		$x(t) = 2X\langle ft + s \rangle - X$ (s = 75% is the phase shift for the shape shown)	$x_{\text{RMS}} = \frac{\sqrt{3}}{3}X$
Three-level modified square wave		$x(t) = \begin{cases} 0, & \langle ft \rangle < 0.25 \\ X, & 0.25 < \langle ft \rangle < 0.5 \\ 0, & 0.5 < \langle ft \rangle < 0.75 \\ -X, & \langle ft \rangle > 0.75 \end{cases}$	$x_{\text{RMS}} = \frac{\sqrt{2}}{2}X$

Note: Peak amplitude is X and mean is zero for all waveforms, except for the DC signal and pulse train, for which average values are nonzero.



$x{\downarrow}\text{PK–PK} = 2x{\downarrow}\text{PK}$

$x{\downarrow}\text{RMS} = \sqrt{2}/2\,x{\downarrow}\text{PK} \cong 0.707x{\downarrow}\text{PK}$

$x{\downarrow}\text{abs} = 2/\pi\, x{\downarrow}\text{PK} \cong 0.637x{\downarrow}\text{PK}$

FIGURE 5.12
Summary of the relationships between average, peak, peak-to-peak, and RMS for harmonic sine waves (these relationships are not valid for nonharmonic signals).

When the time signal is of the harmonic type, the determination of the frequency is a simple process. However, in many applications, time signals are complex functions that combine many different frequencies together as seen in the oscillation of bridges, in speech, or in modulated radio waves. Usually, complex time signals are very hard to interpret and understand; however, when they are represented in the frequency domain, it is often easier to identify some particular features contained in the signal. The Fourier Transform, introduced in 1822 by Joseph Fourier [7], is a mathematical transformation in which a complex function, whether continuous or discontinuous, is expanded into a series of sine waves. Although this result is not absolutely correct, this became among the most widely used tools for transforming a signal in the time domain to the frequency domain. A practical example is shown in Figure 5.13. Applications of the Fourier transform range from designing filters for noise reduction in audio signals to condition monitoring of rotating machinery.

To transform the signal from the time domain to the frequency domain, the Fourier transform decomposes the complex time signal into a series of harmonic time signals. Each individual harmonic component has its own amplitude and frequency, which are plotted in a *xy* axis chart with amplitude on the *y* axis and frequency on the *x* axis. This process is represented schematically in Figure 5.14. If we take the individual waveforms spaced by the frequency (top plane on a parallelepiped), the time and frequency domains

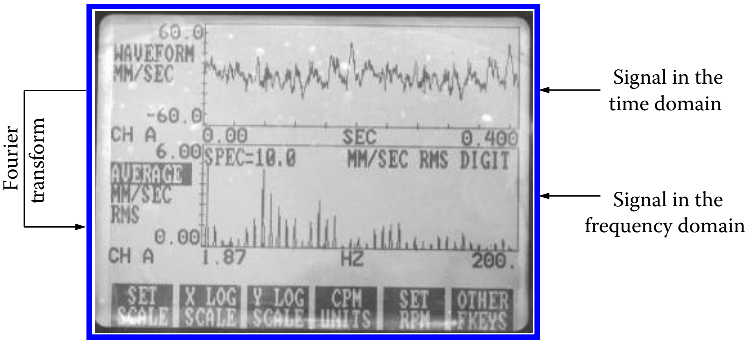

60.0
WAVEFORM
MM/SEC
-60.0
CH A
0.00
SEC
0.400
6.00
SPEC=10.0
MM/SEC RMS DIGIT
AVERAGE
MM/SEC
RMS
0.00
CH A
1.87
HZ
200.
SET SCALE
X LOG SCALE
Y LOG SCALE
CPM UNITS
SET RPM
OTHER FKEYS
Fourier transform
Signal in the time domain
Signal in the frequency domain

FIGURE 5.13
Photo of the screen of an Emerson CSI 2120-2 Machinery Health Analyzer during the measurement of structural vibrations.

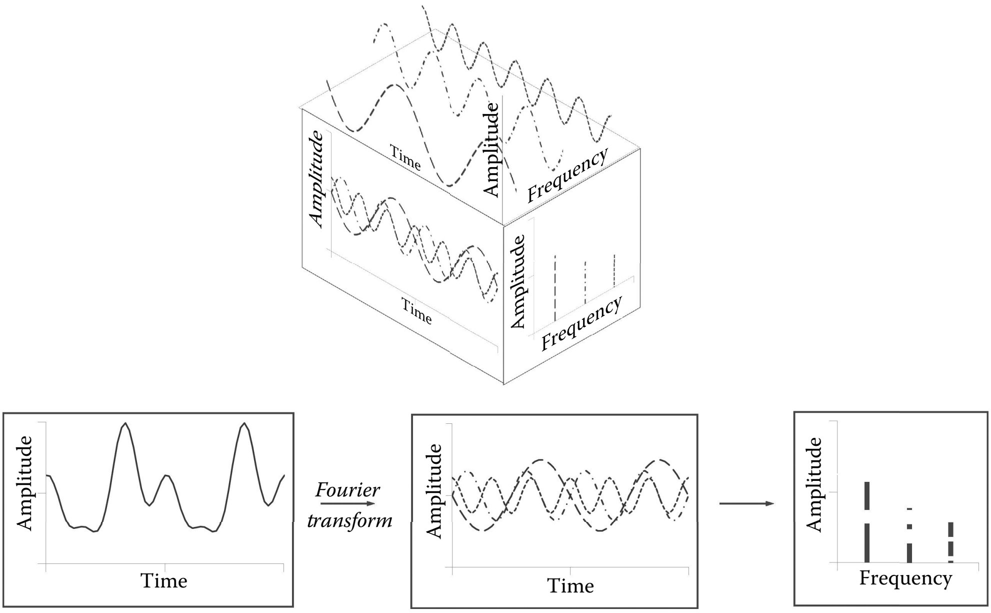

Amplitude
Time
Amplitude
Frequency
Amplitude
Time
Amplitude
Frequency
Amplitude
Time
Fourier transform
Amplitude
Time
Amplitude
Frequency

FIGURE 5.14
Schematic representation of the Fourier transform application on a complex time signal and its representation in both the time and frequency domains.

are the orthographic projections on the other two mutually perpendicular faces of the parallelepiped.

This is an important result because what Figure 5.14 is showing is that each line in the frequency spectrum is a harmonic signal in itself with amplitude and frequency (and phase as well). As a consequence, Equations 5.14, 5.15, 5.18, 5.19, 5.24, and 5.27 and the relationships shown in Table 5.1 and Figures 5.5, 5.7, and 5.12, which were said to be valid for harmonic time signals only when in the time domain, are also valid for complex signals, whether deterministic or not, as long as they are represented in the frequency domain. As an example, these relationships cannot be applied directly on the time signal shown on the top of Figure 5.13, but they can be applied to the frequency spectrum shown below because each line alone represents an individual sine wave.

In this section, the basic concepts of Fourier analysis and Fourier transform are presented. The discrete Fourier transform (DFT) and fast Fourier transform (FFT) will be discussed within this chapter. Their limitations and how they can be implemented will also be addressed.

5.3.2 Fourier Transform

Let us assume the simpler case of a periodic time signal $x(t)$ with period T (deterministic, but not necessarily pure harmonic). First of all, even if the time signal is acquired over a finite period of time, Fourier analysis assumes that the domain of $x(t)$ is infinite, that is, $t \in [-\infty, +\infty]$. As mentioned earlier, Fourier analysis also assumes that this time signal can be represented as a series of a number $n = 1,2,3,\ldots$ of related sinusoidal waves:

$$x(t) = a_0 + 2\sum_{n=1}^{\infty}\left(a_n \cos\frac{2\pi n t}{T} + b_n \sin\frac{2\pi n t}{T}\right) \tag{5.31}$$

Because the domain of $x(t)$ is infinite, we may assume that $t \in [-T/2, +T/2]$. during a complete period of time T. Coefficients a_0, a_n, and b_n are real constants that can be determined from, respectively,

$$a_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\,\mathrm{d}t \tag{5.32}$$

$$a_n = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\cos\frac{2\pi n t}{T}\,\mathrm{d}t \tag{5.33}$$

$$b_n = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\sin\frac{2\pi n t}{T}\,\mathrm{d}t \tag{5.34}$$

in which it can be observed that a_0 simply is the mean value of $x(t)$ over the period T. Coefficients a_n and b_n are called the Fourier, or spectral, coefficients [8].

If exponential complex functions are used instead of trigonometric ones, that is,

$$\cos\frac{2\pi nt}{T}=\frac{1}{2}\left(e^{i\frac{2\pi nt}{T}}+e^{-i\frac{2\pi nt}{T}}\right) \tag{5.35}$$

$$\sin\frac{2\pi nt}{T}=\frac{1}{2i}\left(e^{i\frac{2\pi nt}{T}}-e^{-i\frac{2\pi nt}{T}}\right) \tag{5.36}$$

the resulting exponential form for the Fourier series can be written as

$$x(t)=\sum_{n=-\infty}^{+\infty}c_n e^{i\frac{2\pi nt}{T}} \tag{5.37}$$

where

$$c_n=\frac{1}{T}\int_{-T/2}^{T/2}x(t)e^{-i\frac{2\pi nt}{T}}\,\mathrm{d}t \tag{5.38}$$

and the following relationship holds between the Fourier coefficients:

$$|c_n|=\sqrt{a_n^2+b_n^2} \tag{5.39}$$

It should be noted that, although Equation 5.31 is defined for positive frequencies only, Equation 5.37 is defined for both positive and negative frequencies (e.g., Ref. [9]).

If $x(t)$ is not a periodic function, say a single impulse or a transient, the Fourier theorem is still valid because these types of signals can be seen as periodic signals with period $T = \infty$. Substituting Equation 5.38 into Equation 5.37 yields

$$x(t)=\sum_{n=-\infty}^{+\infty}\frac{1}{T}\left(\int_{-T/2}^{T/2}x(t)e^{-i\frac{2\pi nt}{T}}\,\mathrm{d}t\right)e^{i\frac{2\pi nt}{T}} \tag{5.40}$$

Because $T \to \infty$, $1/T \to 0$, which is the same as saying $1/T = \mathrm{d}f$. On the other hand, the quantity $n/T = n\mathrm{d}f \to f$ and the sum in Equation 5.40 turns into the following integral:

$$x(t)=\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty}x(t)e^{-i2\pi ft}\,\mathrm{d}t\right)e^{i2\pi ft}\,\mathrm{d}f \tag{5.41}$$

This equation can be written in the more convenient form

$$x(t)=\int_{-\infty}^{+\infty}X(f)e^{i2\pi ft}\,\mathrm{d}f \tag{5.42}$$

with

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-i2\pi ft}\,\mathrm{d}t \tag{5.43}$$

Equations 5.42 and 5.43 constitute what is known as the pair of Fourier integrals, defined in from $-\infty$ to $+\infty$. It is this pair that allows transforming a signal from the time domain into the frequency domain as graphically represented earlier in Figure 5.14.

It is also interesting to note that the coefficient c_n expressed by Equation 5.38 is equivalent to Equation 5.43 except it is defined from $-T/2$ to $+T/2$.

5.3.3 Fourier Transform Application Example

Common waveforms were previously shown in Table 5.1. However, these were defined from 0 to $+\infty$ in the time domain. When applying the Fourier transform, it is more convenient if the functions are defined from $-T/2$ to $+T/2$ instead (Table 5.2).

Table 5.2 also includes a column about the waveform's symmetry about the Cartesian system of coordinates. One useful aspect to know when applying the Fourier transform is that some of the Fourier coefficients a_0, a_n, and b_n given by Equations 5.32, 5.33, and 5.34 can be zero, depending on whether the function is odd or even. For odd functions, such as the sine, square, or sawtooth waves, a_0 and a_n are zero and do not need to be evaluated whereas for even functions, such as the cosine or the triangular waves, b_n is the one that is zero and does not need to be evaluated.

Let us assume that a switch periodically changes from on to off and that its behavior can be approximated by the square wave shown in Figure 5.15. We want to evaluate the Fourier transform of this signal up to the eighth component and plot it both in the time domain and frequency spectrum.

The first step consists of determining the fundamental frequency from the direct measurement of the time period $f_0 = \dfrac{1}{T} = \dfrac{1}{0.04} = 25$ Hz.

Now we are in the condition of evaluating if the function is odd or even. The most convenient way of graphically representing a period of this wave is in the period ranging from $-T/2$ to $+T/2$. This is done in Figure 5.16. In this case, the square wave symmetry is even, contrary to what is shown in Table 5.2, because one of the Fourier coefficients is a DC signal (a_0), which corresponds to an offset in the y-axis.

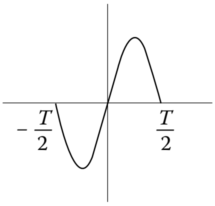
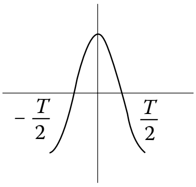
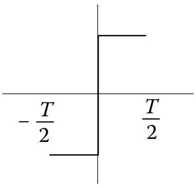
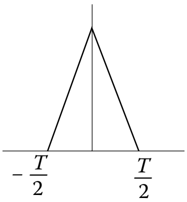
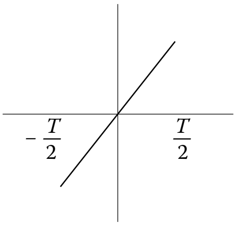
The equation for the square wave shown in Figure 5.16 is

$$x(t) = \begin{cases} 0, -0.02s < t < -0.01s; 0.01s < t < 0.02s \\ 1, -0.01s \le t \le 0.01s \end{cases} \tag{5.44}$$

Now we are in conditions of determining the Fourier coefficients. First of all, because the function represented in Figure 5.16 is even, we know that the Fourier coefficient $b_n = 0$. The Fourier coefficient a_0 is the mean, which, in this case, obviously is $a_0 = 0.5$. This is the DC component of the Fourier spectrum, and it corresponds to an offset of the time signal in the y-axis.

TABLE 5.2

Particular Cases of Common Waveforms and Their Shapes Defined between $-T/2$ and $+T/2$

Waveform	Shape	Equation	Symmetry
Sine wave (sin)	 $-\frac{T}{2}$ $\frac{T}{2}$	$x(t) = X\sin(\omega t)$	Odd $(a_0, a_n = 0)$
Sine wave (cos)	 $-\frac{T}{2}$ $\frac{T}{2}$	$x(t) = X\cos(\omega t)$	Even $(b_n = 0)$
Square wave	 $-\frac{T}{2}$ $\frac{T}{2}$	$x(t) = \begin{cases} -X, & -T/2 < ft < 0 \\ -X, & 0 < ft < T/2 \end{cases}$	Odd $(a_0, a_n = 0)$
Triangular wave	 $-\frac{T}{2}$ $\frac{T}{2}$	$x(t) = \lvert 2Xft - X \rvert$	Even $(b_n = 0)$
Sawtooth wave	 $-\frac{T}{2}$ $\frac{T}{2}$	$x(t) = 2Xft$	Odd $(a_0, a_n = 0)$

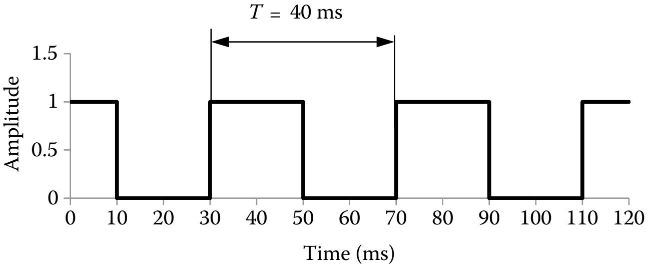


FIGURE 5.15
Example of a square wave with amplitude 1 and fundamental frequency $f_0 = 25$ Hz.

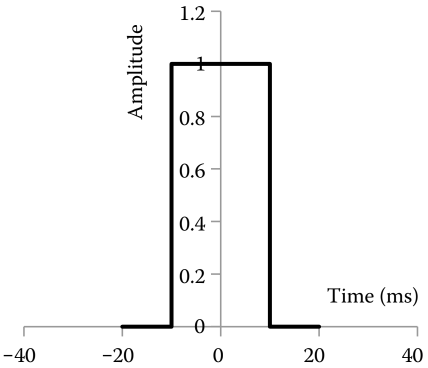


FIGURE 5.16
Example of a square wave with amplitude 1 and fundamental frequency $f_0 = 25$ Hz represented in the $-T/2$ to $+T/2$ domain.

Evaluation of the Fourier coefficient a_n is made with Equation 5.33, which, in this particular case, is

$$a_n = \frac{1}{0.04}\int_{-0.02}^{0.02} x(t)\cos\frac{2\pi nt}{0.04}\,\mathrm{d}t \tag{5.45}$$

This equation can further be simplified if we notice that $x(t) = 0$ for $-0.02\text{s} < t < -0.01\text{s}$ and $0.01\text{s} < t < 0.02\text{s}$. This means that the interval for integration can be reduced to $-0.01\text{s} \le t \le 0.01\text{s}$ where $x(t) = 1$:

$$a_n = \frac{1}{0.04}\int_{-0.01}^{0.01} \cos\frac{2\pi nt}{0.04}\,\mathrm{d}t \tag{5.46}$$

Because the evaluation of the Fourier coefficients require integration, sometimes it is practical to have in mind the following antiderivatives:

$$\begin{aligned}
&\int t\cos at\,\mathrm{d}t = \frac{1}{a^2}\cos at + \frac{1}{a}t\sin at\\
&\int \cos at\,\mathrm{d}t = \frac{1}{a}\sin at
\end{aligned} \tag{5.47}$$

$$\begin{aligned}
&\int t\sin at\,\mathrm{d}t = \frac{1}{a^2}\sin at - \frac{1}{a}t\cos at\\
&\int \sin at\,\mathrm{d}t = -\frac{1}{a}\cos at
\end{aligned} \tag{5.48}$$

The Fourier coefficients a_1 to a_8 are evaluated for $n = 1\ldots8$, respectively. As an example, the third coefficient is determined below:

$$a_3 = \frac{1}{0.04}\int_{-0.01}^{0.01} \cos\frac{3\times 2\pi t}{0.04}\,\mathrm{d}t = \frac{1}{0.04}\left[\frac{0.04}{6\pi}\sin\frac{6\pi t}{0.04}\right]_{-0.01}^{0.01}$$
$$= \frac{1}{6\pi}\left(\sin\frac{6\pi\times 0.01}{0.04} - \sin\frac{6\pi\times -0.01}{0.04}\right) = -0.1061 \tag{5.49}$$

The frequency associated to coefficient a_3 is $f_3 = 3 \times f_0 = 75$ Hz.

After following the same procedure for all the other seven coefficients, the following results should be obtained: $a_1 = 0.3183$; $a_3 = -0.1061$; $a_5 = 0.0637$; $a_7 = -0.04547$; and a_2, a_4, a_6, $a_8 = 0$. Once the Fourier transform has been applied, the time signal can be written as in Equation 5.31, which, in this particular case, takes the form

$$\begin{aligned} x(t) = 0.5 &+ 0.637\cos(2\pi\times 25t) - 0.212\cos(2\pi\times 75t) \\ &+ 0.127\cos(2\pi\times 125t) - 0.0909\cos(2\pi\times 175t) \end{aligned} \tag{5.50}$$

Equation 5.50 is an approximation of what is shown in Figure 5.15, accomplished by adding up harmonic time signals, each one with amplitude, frequency, and phase. A summary of the results is presented in Table 5.3. The graphical representation of this approximated square wave, in both the time and frequency domains, is presented in Figures 5.17 and 5.18, respectively.

TABLE 5.3

Summary of the Results of the Fourier Transform Applied to the Wave in Figure 5.15

Frequency (Hz)	Amplitude (V)	Phase (°)
0 (DC)	0.5	–
25	0.637	0
75	0.212	180
125	0.127	0
175	0.0909	180

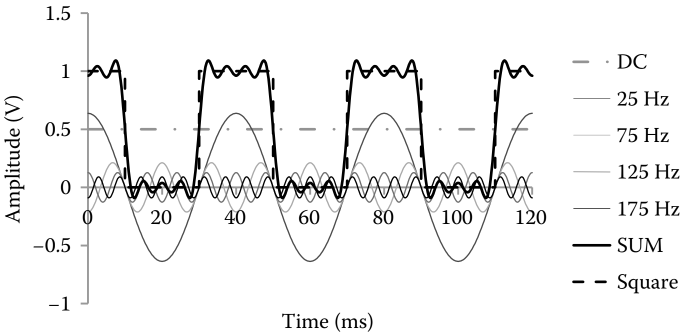


FIGURE 5.17
Time domain representation of the square wave shown in Figure 5.15 after applying the Fourier transform with $n = 8$.

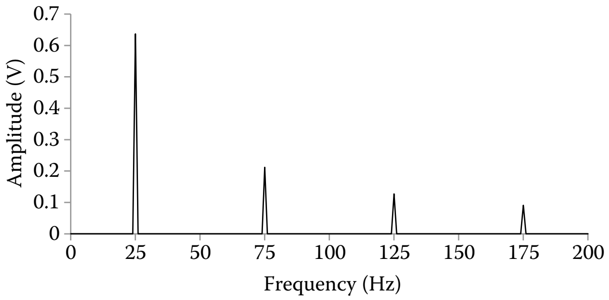


FIGURE 5.18
Frequency spectrum of the square wave shown in Figure 5.15 after applying the Fourier transform with $n = 8$.

The "SUM" wave shown in Figure 5.17 is the total of four pure harmonic sine waves at related frequencies, amplitudes, and phases. If more components would have been considered (i.e., Fourier coefficients), it would be expected that the "SUM" wave would better approximate the original square one.

Nevertheless, when the original signal is a discontinuous function—which is the case of the square wave discussed—the Fourier transform will not be able to perfectly regenerate the signal without some overshoot. This is often called the Gibbs phenomenon and is depicted in Figure 5.17 at the "corners" of the wave [10].

5.3.4 Basics of the Discrete and Fast Fourier Transforms

5.3.4.1 Sampling Frequency

Real-world waveforms are continuous functions of time: Signals are analogic as well as most sensors' output. For example, a preamplified piezoelectric accelerometer (see Section 5.5.1) produces an output signal that is proportional to the acceleration of the system at the measurement coordinate. However, in signal processing, there is always a process of ADC at some point, which, in this case, consists of converting the analog transducer signal into the digital code used by the processor [6]. Figure 5.19 shows a signal acquisition module from National Instruments that includes conditioners. The input signals it receives are continuous analog time signals from transducers, which are converted through an electronic ADC into a discrete time series digital signal.

The ADC converter records the level of the signal at a discrete set of times. Figure 5.20 illustrates the sampled acquisition of an analog time signal, where $\Delta = 1/f_s$ is the time elapsed between each sample, and f_s is the sampling frequency. In the example case of the National Instruments acquisition module shown in Figure 5.19, the sampling frequency is 51.2 kS/s (kilo-samples per second) per channel [11]. In practice, what is available is a collection of points with information on the amplitude at discrete and regular intervals of times.

The signal acquisition module shown in Figure 5.19 has separate conditioners for each one of the four channels. Less expensive alternatives time-share the signal with multiplexers. Multiplexers will pick one channel of data from a bank of data channels in a sequential manner and connect it to a common input device. Basically, it distributes the sampling frequency through the bank of channels. In a multiplexer, each channel in the bank will have a delay proportional to the inverse of the sampling frequency.

The minimum sampling frequency f_s required in ADC must be at least twice as much the signal's fundamental frequency f_0, that is, $f_s/f_0 > 2$. This means that with a sampling

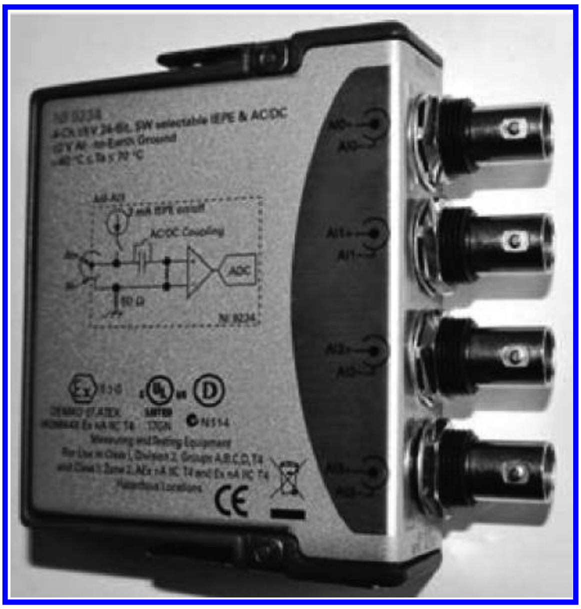


FIGURE 5.19
National Instruments four-channel C Series dynamic signal acquisition module (NI 9234) for audio/frequency measurements from integrated electronic piezoelectric (IEPE) and non-IEPE sensors with NI CompactDAQ or CompactRIO systems (NI 9234, available at http://sine.ni.com/nips/cds/view/p/lang/en/nid/208802).

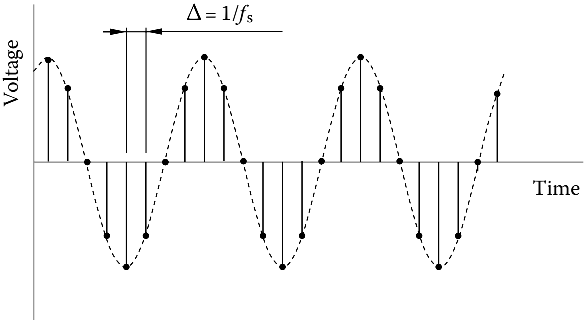


FIGURE 5.20
Sampled acquisition of a voltage analog time signal.

frequency of 51.2 kS/s, the NI 9234 shown in Figure 5.19 is adequate for waveform measurements up to 25.6 kHz. This is a very important result in signal processing that will be later addressed in Section 5.4.1, when discussing problems associated with signal processing, namely aliasing.

5.3.4.2 Discrete Fourier Transform

Loosely speaking, when a continuous function is sampled at discrete and regular intervals of time, the major difference in the post-processing mathematics is that integrals are expressed in terms of sums, which is not far from the definition of the Riemann integral. So if $x(t)$ is sampled at regular intervals of time and represented by the discrete series

$\{x(k)\}$, $k = 0,1,2,\ldots, N-1$ where $t = kT/N$, it can be shown that the pair of Fourier integrals (Equations 5.42 and 5.43) can be replaced by the summations [6]

$$x(k) = \sum_{j=0}^{N-1} X(j)e^{i2\pi jk/N} \tag{5.51}$$

with

$$X(j) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)e^{-i2\pi jk/N} \tag{5.52}$$

and $j = 0,1,2,\ldots, N-1$.

These summations constitute the DFT pair. In practice, the computation of the Fourier transform by computers and spectrum analyzers is done after the ADC, which means that this is made over a discrete time series. However, computation of Equation 5.52 would require nearly N^2 complex operations. For large values of N this can become computationally prohibitive even for modern and fast processors.

5.3.4.3 Fast Fourier Transform

The fast Fourier transform (FFT), as the name suggests, is a much faster algorithm based on the DFT that has become very popular since its development in the 1960s [12,13]. It is not the intention of this book to describe its details or the ones from the many derivations subsequently developed. However, an important result of the use of the FFT that affects the sampling frequency will be addressed. This is important because it gives a better understanding of how spectrum analyzers operate.

It has been found that the number of operations in the FFT can be reduced from nearly N^2 to $(N/2)\log_2 N$ multiplications, $(N/2)\log_2 N$ additions, and $(N/2)\log_2 N$ subtractions for the more general case of $N = 2^m$ where m is any positive integer [6]. This is illustrated in the plot shown in Figure 5.21, in which the computational benefits of the use of the FFT instead of the DFT become quite obvious, especially for larger values of N.

Because $N = 2^m$, where m is any positive integer, this means that the number of points that are acquired over time can be from 2 to 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,

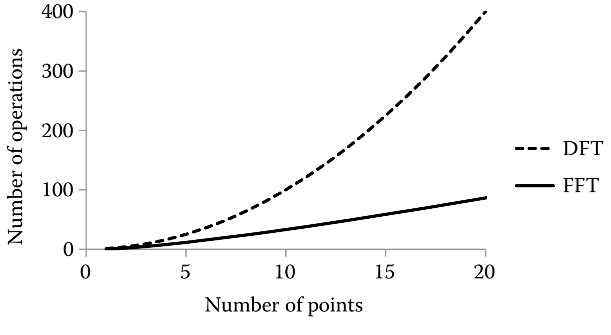


FIGURE 5.21
Contrast between the number of operations needed for evaluating the DFT and the FFT.

9192, etc. The time T that is required to acquire a certain amount N of points depends on the sampling frequency f_s:

$$N = Tf_s \tag{5.53}$$

Now, let us suppose the following problem: A time signal is meant to be represented in the frequency domain up to $f_{max} = 400$ Hz with a total of $n_{lines} = 1600$ spectral lines. The frequency resolution in the frequency spectrum is determined from

$$f_{res} = \frac{f_{max}}{n_{lines}} \tag{5.54}$$

Thus, in this example, $f_{res} = 0.25$ Hz. The duration for the acquisition of a time period is $T = 1/f_{res} = 4$s. Because the measurement is being made up to 400 Hz, this means that the minimum sampling frequency must be at least $f_s = 800$ Hz to avoid aliasing, according to the Nyquist–Shannon theorem that will be later described in Section 5.4.1. So the number of points needed for the FFT, according to Equation 5.53, is $N = 4 \times 800 = 3200$ points. However, this number of points is not a power of two and cannot be used in the FFT algorithm. The closest powers of two to 3200 are $2^{11} = 2048$ and $2^{12} = 4096$. Thus, we must maximize the number of points to 4096 to include all 3200 points required.

Now it is necessary to check if the Nyquist–Shannon theorem is being respected with this new value $N' = 4096$. Equation 5.53 can be used once more to determine the new sampling frequency: $f_s' = 4096/4 = 1024$ Hz. The ratio between this new 1024-Hz sampling frequency and the 800-Hz original one of is 2.56, which means that the spectrum analyzer is giving information up to $0.39 f_s$.

Most spectrum analyzers frequently show a number of lines n_{lines} that is based on 25×2^m, namely 25, 50, 100, 200, 400, 800, 1600, etc. However, even if they do not, it is easy to understand that the sampling frequency in the FFT will always be larger than two and thus obey the Nyquist–Shannon theorem because the number of sampling points in the time domain must be majored by the following power of two.

5.4 Signal Processing

5.4.1 Aliasing

On an ADC as the one shown in Figure 5.20, information is lost about the time periods that elapse between each sample of the signal, and one assumes what it is in between by extrapolation. As a consequence, if the sampling frequency is not chosen correctly, results may be misleading. Figures 5.22 and 5.23 give a few examples of what may happen when a sine wave is sampled at four different sampling frequencies, in both the time and frequency domains.

Let us assume a continuous sinusoidal harmonic signal with frequency $f_0 = 100$ Hz and amplitude 1 V is being sampled at a frequency $f_s = 100$ Hz. The ratio between the sampling frequency and the signal frequency is $f_s/f_0 = 1$. The resulting data misleads us to think the signal is a DC value. This conclusion is wrong because the original signal is a sine wave. Furthermore, the amplitude of the DC signal so obtained in this particular example is 0.5 V whereas the amplitude of the true sine wave is twice as much.

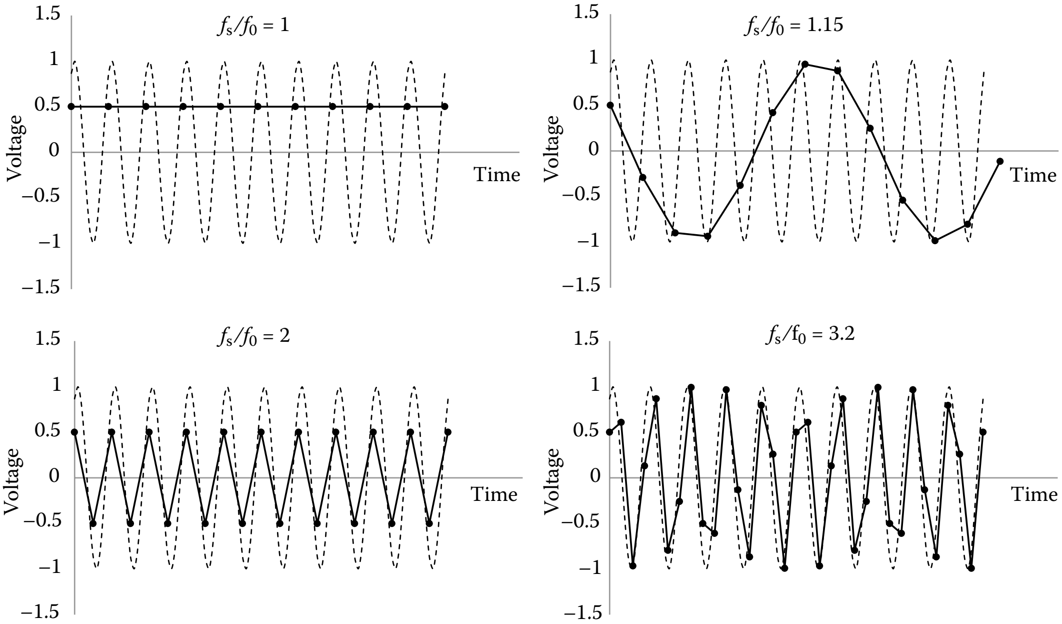


FIGURE 5.22
A sine wave is sampled at different ratios between sampling frequency and signal's frequency—representation in the time domain.

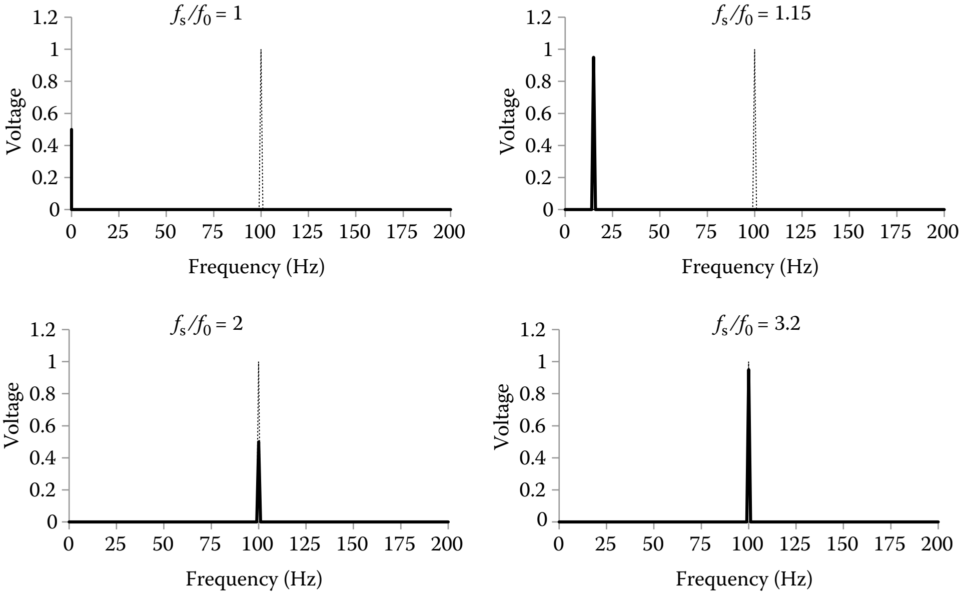


FIGURE 5.23
A sine wave is sampled at different ratios between sampling frequency and signal frequency—representation in the frequency domain.

Let us now assume that the sampling frequency is increased to $f_s = 115$ Hz so that the ratio between the sampling frequency and the wave frequency is $f_s/f_0 = 1.15$. In this case, Figure 5.22 shows a sine wave with an amplitude near to 1 V but at a much lower frequency than the true signal. Actually, the frequency of this sine wave is the difference between the sampling frequency and the signal's frequency $f_s - f_0 = 15$ Hz. The misinterpretation of a signal by another one at a lower frequency due to undersampling is called *aliasing.**

In the third case shown in Figure 5.22, the sampling frequency is twice as much as the signal frequency, that is, $f_s = 200$ Hz and $f_s/f_0 = 2$. The sampled data now shows a triangular wave with 100 Hz although the amplitude is still misleading. The amplitude of the sampled data depends on when the first sample was taken (phase). This occurs whenever the wave is sampled at a ratio that is an integer fraction of its frequency ($f_0/f_s = 1,2,3,\ldots$). Thus, it is reasonable to assume that for $f_s/f_0 > 2$ it will be possible to measure the amplitude. This is known as the Nyquist–Shannon theorem, in which it is stated that the minimum sampling rate required for aliasing to be avoided must be at least twice as much the signal's frequency.

Finally, in the case in which the ratio between the sampling frequency and the signal frequency is larger than 2 ($f_s/f_0 = 3.2$), it is possible to measure both the frequency and amplitude even if the signal is oscillating and only occasionally reaches the peak amplitude of 1 V.

In real measurements, the sampling frequency is set according to the highest value in the frequency range of interest. Let us assume a measurement is being carried out in the 0 to 800 Hz frequency range. The highest frequency in the range is the one that needs the faster sampling speed, that is, it is the one frequency that determines the sampling frequency. Thus, in this case and according to the previous discussion, the sampling frequency would have to be set to at least $f_s = 2 \times 800 = 1600$ Hz in order to avoid aliasing.

We now ask the question: is this enough to prevent aliasing? To answer this question, let us assume that we want to plot the frequency spectrum of a system (initially unknown to us) that is producing a signal with the following components: 300 Hz, 600 Hz, 900 Hz, and 1200 Hz. If we now evaluate the ratios between the sampling frequency and these components' frequencies, for $f_s = 1600$ Hz, we notice that the Nyquist–Shannon theorem is not respected at all the available frequencies: $\frac{1600}{300} \cong 5.3$, $\frac{1600}{600} \cong 2.7$, $\frac{1600}{900} \cong 1.8$, and $\frac{1600}{1200} \cong 1.3$. In this case, the ratios between the 900 and 1200 Hz components of this signal and the sampling frequency are both smaller than two, which mean that aliasing will occur. This is illustrated in Figure 5.24, in which two inexistent signals, called "aliases," will appear on a frequency spectrum analyzer's screen at 400 and 700 Hz. The alias peak at 400 Hz corresponds to the 1200 Hz component (1600 − 1200 = 400 Hz), and the alias peak at 700 Hz corresponds to the 900 Hz component (1600 − 900 = 700 Hz).

This means that, per se, the sampling frequency is not enough to avoid aliasing unless it is known beforehand that there are no higher frequencies in the signal, which, most of the times, is not possible to predict.

Combined with the correct sampling frequency setting, it is necessary to filter the signal. However, it must be stressed that digital filtering does not avoid aliasing because if

---

* One practical example in which aliasing is seen is when watching a film of a moving car: Sometimes there is the illusion that the wheels are spinning contrary to the direction of motion; other times the wheels seem to be rotating at a much lower speed than one needed for the car to move at a certain velocity. This is because cameras also have sampling frequency. Actually, a film is a composition of a sequence of photos, which is equivalent to the digital sampling process in a waveform. Typical frame rates for cameras today may range from 24 to 300 fps (frames per second).

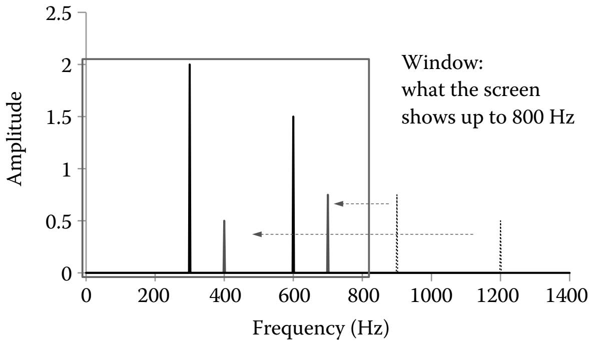
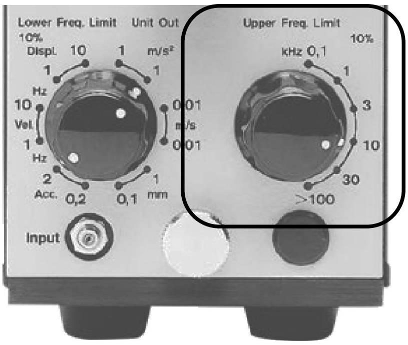


FIGURE 5.24
Example of aliasing on the acquisition of a signal with four harmonic components (300 Hz, 600 Hz, 900 Hz, and 1200 Hz) when measured up to 800 Hz with a sampling frequency of 1600 Hz (no filtering).

the signal has already been digitized, sampling has already been done. Filtering must be done on the analog signal prior to sampling by the hardware. Most modern acquisition systems and ADCs, such as the one shown in Figure 5.19, bring built-in conditioners with low-pass filters with variable cutoff frequency based on the sampling frequency. However, older systems may only offer a selection of available cutoff frequencies as is the case of the charge amplifier type 2635 from Brüel & Kjær shown in Figure 5.25.

The charge amplifier is a current integrator that converts the input charge from an electrical source with a capacitive nature (say, a piezoelectric charge accelerometer) into a proportional output voltage. The one shown in Figure 5.25 offers a low-pass filter with the following possible cutoff frequencies: 100 Hz, 1000 Hz, 3000 Hz, 10 kHz, and 30 kHz. Let us now assume that the frequency spectrum shown in Figure 5.24 is being acquired with a charge accelerometer that is connected to the charge amplifier shown in Figure 5.25. The maximum frequency we want to measure, 800 Hz, is between the 100 and 1000 Hz cutoff frequencies of the charge amplifier. If we set the cutoff frequency, say, at 100 Hz, we will not be able to measure anything above 100 Hz, so the cutoff frequency must be set to at least 1000 Hz.

Figure 5.26 shows the frequency spectrum of the original signal shown in Figure 5.24 but after a low-pass filter with cutoff frequency set to 1000 Hz has been applied to the analog signal (before digitalization). It is possible to observe that the component at 1200 Hz does



Low-pass filter
cutoff frequency
selection knob

FIGURE 5.25
Charge amplifier type 2635 from Brüel & Kjær.

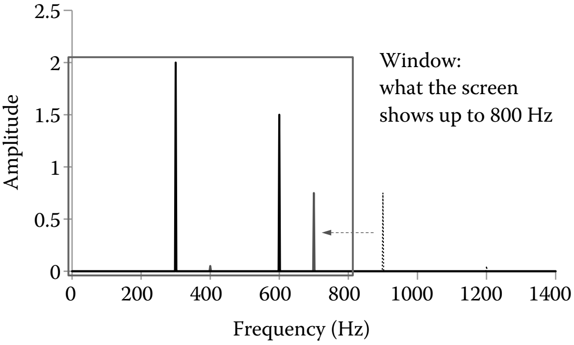


FIGURE 5.26
Example of aliasing on the acquisition of a signal with four harmonic components (300 Hz, 600 Hz, 900 Hz, and 1200 Hz) when measured up to 800 Hz with a sampling frequency of 1600 Hz (low-pass filter with cutoff frequency set to 1000 Hz).

not produce an alias at 400 Hz anymore because the analog filter eliminated the 1200 Hz component before the signal was sampled (if the filter was not able to eliminate it completely, at least its level has been significantly reduced in comparison to the other components in the spectrum). However, there still exists the alias frequency at 700 Hz. This is because the sampling frequency was not adjusted when the filter was applied. Saying that the filter cutoff frequency is 1000 Hz is equivalent to saying that the ADC will have to deal with a signal up to 1000 Hz even if the screen only shows up to 800 Hz. This means that the Nyquist–Shannon condition for the sampling frequency must be based on the analog filter available. Thus, the sampling frequency should be adjusted to $f_s = 2000$ Hz, which is twice the cutoff frequency of the low-pass analog filter available (Figure 5.27).

In summary, to make sure aliasing is eliminated, two conditions must be met simultaneously:

1. An analog low-pass filter must be used before the signal is sampled in the ADC.
2. The sampling frequency must be set to a value at least twice as much the value of the cutoff frequency set in the analog low-pass filter.

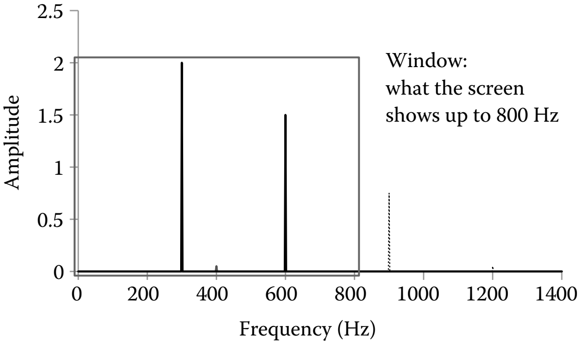


FIGURE 5.27
Example of aliasing on the acquisition of a signal with four harmonic components (300 Hz, 600 Hz, 900 Hz, and 1200 Hz) when measured up to 800 Hz with a sampling frequency of 2000 Hz (twice as much the value of the low-pass cutoff frequency set to 1000 Hz).

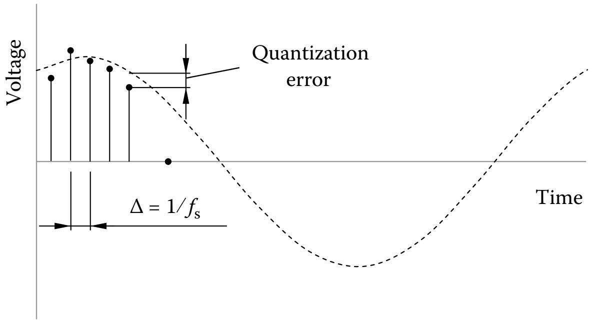


FIGURE 5.28
Example of quantization errors during digital sampling of a signal.

5.4.2 Quantization Errors

When an ADC occurs, there also are a number of possible digital levels for the amplitude, called quantization levels. This means that the level of the signal at the instant it is sampled is rounded to the nearest digital level as shown in Figure 5.28.

The accuracy of the quantization process depends on the number of bits in the converter: 2^{nbit}. For example, the National Instruments signal acquisition module shown in Figure 5.19 has a 24-bit ADC. The number of peak-to-peak quantization levels is $2^{24} = 16777216$ levels (including zero), and the dynamic range is determined to be $20\log_{10} 2^{24} = 144.5$ dB for peak-to-peak measurements.* This is equivalent to 138.5 dB for peak measurements and a 135.5 dB dynamics range for RMS measurements.

It is recommended that the signal occupies as much of the range of the converter as possible. The loss in the dynamic range is $20\log_{10}\dfrac{\text{Signal level}}{\text{Converter range}}$. This means that if the signal is 10 times smaller than the range in the converter, 20 dB of the measurement range will be lost. If the signal is 100 times smaller than the range in the converter, 40 dB of the measurement range will be lost, and so on. Thus, ideally, signals should be amplified to occupy as much as possible the full range of the converter to minimize quantization errors and improve the signal-to-noise ratio. Another possibility, if existent, is to change the range of the converter. This is possible, for example, when using piezoelectric charge accelerometers that require a charge amplifier. Charge amplifiers often include the possibility to change the dynamic range in addition to band-pass filters.

5.4.3 Leakage and Windowing

The Fourier series given by Equations 5.31 or 5.37 shows that $x(t)$ can be represented by a series of harmonic components with frequencies $f_1 = f_0, f_2 = 2f_0, f_3 = 3f_0$, etc., where $f_0 = 1/T_0$ is both the fundamental frequency and the resolution in Hz (spacing of the frequency components).

A Section 5.3.3 example showed that the Fourier analysis works over a function $x(t)$ with period T_0. However, this does not mean that only one period of signal exists; on the contrary, Fourier analysis assumes that signals repeat infinitely in time. If the signals are truly periodic, this is not a problem, because there cannot be any components in the signal at frequencies between those calculated in the Fourier analysis. However, there are many real situations in which this is not the case either because the signal being acquired has a non-integer number of periods during the measurement period or because the signal is of the random type. Actually, unless there is full control over the source signal and perfect synchronization with the acquisition clock, in most practical situations, it is not possible to guarantee an integer number of periods is being acquired.

---

* The dB is the name of a logarithmic scale that will be explained later in Section 5.6.1.

Let us analyze two contrasting situations, shown in Figure 5.29: In situation (a), n integer periods of a sine wave are measured during an integer period of time nT; in situation (b), $n + 0.5$ periods of the same sine wave are measured during a non-integer period of time $(n + 0.5)T$.

Because only a finite portion of the signal can be measured, and the Fourier transform assumes that time signals are periodic and repeat infinitely in time, what the Fourier transform will "see" is what is shown in Figure 5.30 instead.

Figure 5.30b is no longer a harmonic signal; thus, when applying the Fourier transform, the signal shown in Figure 5.30b will be regenerated as a sum of sine waves that will be translated into spectral components in the frequency domain. In practice, what happens is that the fundamental frequency is shown with smaller amplitude with spreading to neighboring frequencies. The phenomenon of spreading of the true spectrum components to other frequencies is called *leakage* and is illustrated in Figure 5.31.

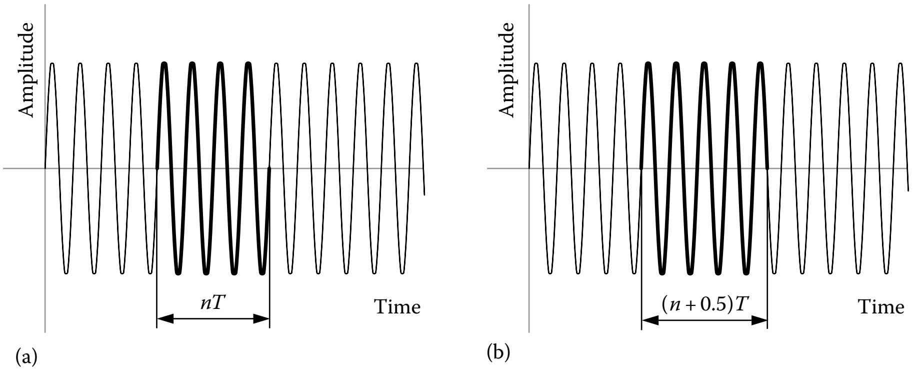


FIGURE 5.29
Acquisition of a sine wave during (a) an integer number of periods and (b) a non-integer number of periods.

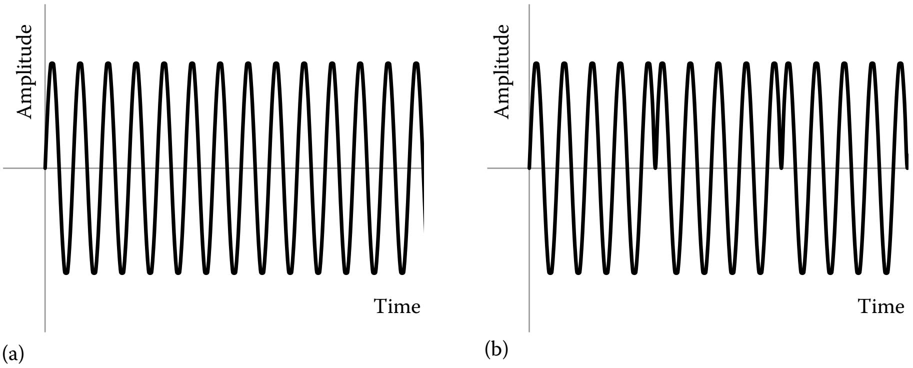


FIGURE 5.30
What the Fourier transform "sees" after the acquisition of the sine waves shown in Figure 5.29: (a) integer number of periods and (b) non-integer number of periods.

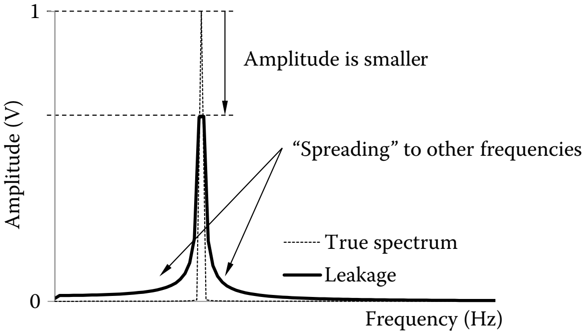


FIGURE 5.31
Overlapping of the true power spectra (dashed line) with the one obtained when the number of periods is a half integer (solid line). No window is applied to the time signal. Leakage occurs and is highly visible; the amplitude is different from its true value and "spreading" of the true spectrum components to other frequencies is accentuated.

This phenomenon can be seen as a distribution of the energy contained in the fundamental spectral line (which is the same in the particular examples shown in Figures 5.29 and 5.30) to contiguous frequencies as if the peak "melted" and "leaked" to the sides.

In real situations, it is very difficult to guarantee that an integer number of cycles is acquired during a measurement even if the signal is periodic. Even in a laboratorial environment, in which control can be taken over the excitation source, this may be impossible to guarantee if the signals are of the random type. To avoid—or at least minimize—the leakage phenomenon, a function, known as a *window,* is multiplied by the time signal before the Fourier transform is performed. The objective is to obtain a smooth decay to zero at the limits of the recorded time period, so that the resulting signal is continuous and approximates more closely a periodic one. Figure 5.32 shows an example of how a window may change a time signal before it is transformed into the frequency domain. In this example, a non-integer number of 16.5 cycles has been measured. It can be seen that, at the center, the windowed time signal has a periodic appearance (solid line) at the location where the original time signal presents a discontinuity (dashed line).

Other applications of the window functions include finite impulse response (FIR) filter design or beam forming because the window itself can be seen as a type of a digital filter.

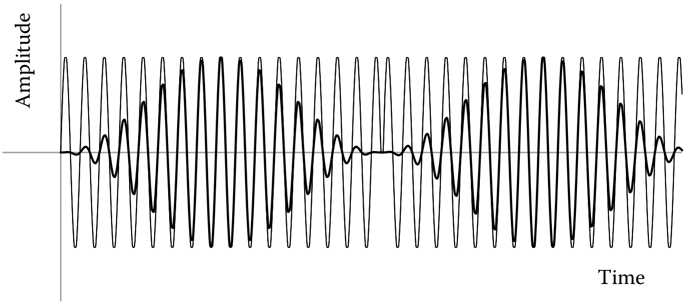


FIGURE 5.32
Representation of two measurement periods of a sinusoidal time signal containing a non-integer number of cycles (16.5). The dashed line is the time signal without the window, and the solid line is the time signal after being multiplied by a window function (Hanning, in the example).

Many windows have been developed as an attempt to reduce as much as possible leakage and other problems related to its usage. Just to name a few, there are uniform (or rectangular), Barlett (or triangular), B-Spline, Welch, Hanning, Hamming, Blackman, Nutall, flat top, Tukey, Rife–Vincent, Gaussian, Kaiser–Bessel, Slepian, Parzen, Bohman, Dolph–Chebyshev, ultraspherical, exponential (or Poisson), etc. There also exist hybrid windows that result from the combination of two windows, either by multiplication or summation. However, it is not the intent of this text to explore in depth how different windows compare, but rather to give an overview on the use of some windows to prevent leakage in typical applications.

In this book, windows were first divided into shapes: rectangular, triangular, bell-shaped, and exponential. These shapes and their generating functions are represented in Table 5.4. In this table, the most widely used bell-shaped windows are represented by a generalized cosine function, the coefficients of which are presented in Table 5.5. However,

TABLE 5.4

Typical Window Functions and Their Shapes

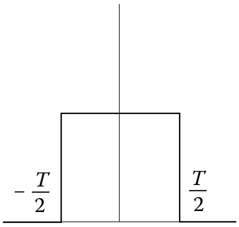
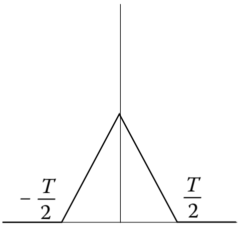
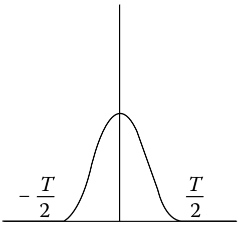
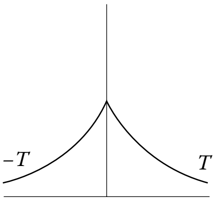
Window	**Shape**	**Equation**
Rectangular or uniform (no window)		$w(t) = 1, \quad \lvert t\rvert \le \frac{T}{2}$ $w(t) = 0, \quad \lvert t\rvert > \frac{T}{2}$
Triangular (Barlett)		$w(t) = 1 - \frac{2\lvert t\rvert}{T}, \quad \lvert t\rvert \le \frac{T}{2}$ $w(t) = 0, \quad \lvert t\rvert > \frac{T}{2}$
Bell-shaped (generalized cosine)		$w(t) = \sum_{k=0}^{n} a_k \cos\frac{2\pi kt}{T}, \quad \lvert t\rvert \le \frac{T}{2}$ $w(t) = 0, \quad \lvert t\rvert > \frac{T}{2}$
Exponential	 Exponential	$w(t) = e^{-\left\lvert t-\frac{T}{2}\right\rvert\frac{1}{\tau}}$ with $\tau = 10\frac{T}{D}\log e$ and D the target decay in dB over half of the window length

TABLE 5.5

Coefficients for Typical Bell-Shaped Window Functions

Window	n	a_0	a_1	a_2	a_3	a_4
Hanning	1	0.5	0.5	–	–	–
Hamming	1	0.54	0.46	–	–	–
Blackman	2	0.427	0.497	0.0768	–	–
Nutall	3	0.356	0.487	0.144	0.0126	
Flat top	4	0.216	0.416	0.278	0.0837	0.00604

it should be noted that not all the bell-shaped functions are generalized cosine functions, for example, the Kaiser–Bessel. On the other hand, some functions, such as the flat top, are not truly bell-shaped because it is partially negative on the sides. In the particular case of the Hamming window, it is not zero at $\frac{T}{2}$.

The most commonly used window function is the Hanning one, which belongs to the bell-shaped category. This window was named after Julius von Hann, a meteorologist who applied an equivalent process to meteorological data. When the Hanning window is applied to the signal presented by the solid line in Figure 5.29b, the spectrum obtained after performing the Fourier transform is the one shown in Figure 5.33. When this spectrum is compared to the one obtained in Figure 5.31, it is possible to observe that the spreading of the fundamental frequency to neighboring spectral frequencies is much less pronounced. Moreover, the amplitude of the signal was made closer to its true value. As such, the use of a window generally reduces the extent of leakage and the chances of important components of the signal being masked by the leaked components.

In addition to the Hanning window, the rectangular and exponential windows are widely used as well. The rectangular or uniform window corresponds to a situation in which no window is used. This can be useful when the signal is truly periodic in time T and sampling is synchronous. The exponential window is useful for transient signals, which had faded away within the time record. This can be particularly useful when measuring impulses or impacts.

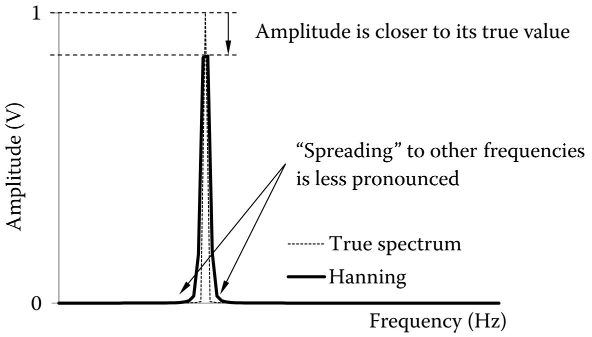


FIGURE 5.33

Overlap of the true power spectrum (dashed line) with the one obtained when the number of periods is a half integer (solid line). A Hanning window is applied to the time signal. Leakage still occurs but is less visible (solid line) than in Figure 5.31. The amplitude is closer to its true value, and "spreading" of the true spectrum components to other frequencies is less pronounced.

In summary, the choice of the window depends on the application. A few recommendations on the choice are below:

1. Hanning: for general application (random signals)
2. Blackman: when spectral leakage must be minimized (random signals)
3. Flat top: for accuracy in amplitude measurements (random signals)
4. Rectangular (or uniform): for periodic signals with synchronous sampling
5. Exponential: for transient signals (generated from impulses)

5.4.4 Convolution

The convolution of two time signals is regarded by some authors as the most important technique in digital signal processing (DSP) [14]. In convolution, systems are described by an IRF.

First, it is important to define what an impulse is. The simplest form of a nonperiodic input function is the unit impulse or Dirac δ-function, represented by

$$f(t) = \delta(t - \tau) \tag{5.55}$$

which is zero for all values of t except for $t = \tau$, where

$$\lim_{\Delta t \to 0} \int_{\tau}^{\tau+\Delta t} f(t)\,\mathrm{d}t = 1 \tag{5.56}$$

The impulse function just defined is represented in Figure 5.34 by a unitary rectangular area of width Δt and height $1/\Delta t$, with $\Delta t \to 0$.

The unit IRF is defined as the response of a system for a unitary impulse of the type represented in Figure 5.34 and is represented as $h(t - \tau)$. In other words, this function is the transfer function of the system for a unitary impulse input (Figure 5.35).

For linear systems, the principle of superposition applies. Linear systems are those in which the response is proportional to the excitation. For example, in single input multiple output (SIMO) systems, if a given input is doubled, the resulting outputs are doubled as well (considering the system is linear). In MIMO systems, the response due to two simultaneous inputs is equal to the sum of the responses when the inputs are applied one at a time.

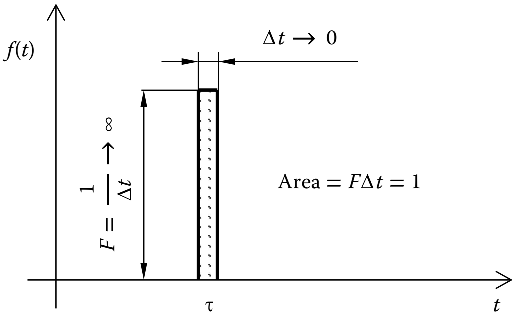


FIGURE 5.34
Definition of a unit impulse forcing function represented by a rectangular area of width Δt and height $1/\Delta t$, with $\Delta t \to 0$.

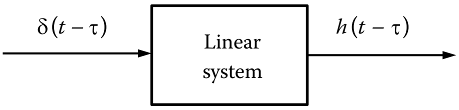


FIGURE 5.35
IRF of a linear system is the output of the system when the input is a Dirac δ-function.

Thus, for linear systems, the response to an arbitrary input function $f(t)$ may be taken as the superposition (sum or integral) of the responses to a series of impulses that, when summed up, represent the input function. In this case, the output function can be represented as

$$x(t) = \sum_{\tau} f(\tau)h(t-\tau)\Delta\tau \tag{5.57}$$

which, when taken to the limit as $\Delta\tau$ tends to zero, becomes

$$x(t) = \int_{-\infty}^{t} f(\tau)h(t-\tau)\,\mathrm{d}\tau \tag{5.58}$$

This integral is called the convolution or Duhamel's integral [6]. Substituting variable τ by $\tau = t - \kappa$, $\mathrm{d}\tau$, becomes $-\mathrm{d}\kappa$, and the limits of integration change. As a consequence,

$$x(t) = \int_{0}^{+\infty} f(t-\kappa)h(\kappa)\,\mathrm{d}\kappa \tag{5.59}$$

This expression, in which $h(\kappa)$ plays a role of weighting function (also called memory) [15], is considerably simplified by taking the Fourier transform, which yields

$$X(\omega) = H(\omega)F(\omega) \tag{5.60}$$

where $H(\omega)$ is the Fourier transform of the IRF. This is a very important result: First, convolutions transform to products [15]; second, it is possible to derive the frequency response function (FRF)* for a given system just by taking the Fourier transform of the IRF [6].

The convolution can be seen as an operation between two signals as a sum or a multiplication is. This is often represented with a*. In terms of applications, it can be used to solve many mathematical problems, ranging from statistics to differential equations, including DSP [14].

5.4.5 Random Signals

5.4.5.1 Auto-Spectrum, Power Spectrum, and Cross-Spectrum

In many real applications, signals are not deterministic. As such, random signals cannot be treated the same way as deterministic signals are; that is, it is not actually correct to analyze them assuming a periodicity of infinite period [6]. This is because the Dirichlet condition cannot be satisfied for infinite random signals, that is,

---

* The FRF—explained later—is a transfer function relating at least one output to one input (for example, the vibration response $x(t)$ to a given random excitation force $f(t)$).

$$\int_{-\infty}^{+\infty} \left| x(t) \right| \mathrm{d}t < \infty \tag{5.61}$$

This condition must be satisfied for the Fourier transform to be used. Thus, the analysis of random signals is made using statistical approaches instead, such as the *auto-correlation* function. The auto-correlation of an ergodic* signal $x(t)$ is the correlation of that signal with itself when measured after a time lag τ, $x(t + \tau)$:

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)x(t+\tau)\,\mathrm{d}t \tag{5.62}$$

For a discrete signal, it takes the form

$$R_{xx}(\delta) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x_n x_{n+\delta} \mathrm{d}t \tag{5.63}$$

This mathematical formulation allows finding repeating patterns in a signal, such as a periodic signal that is invisible when covered by excessive levels of noise.

To better understand what the auto-correlation function looks like, let us take as an example a 10-Hz harmonic sine wave with added random white noise:

$$x(t) = \sin(20\pi + \pi/3) + \textit{random noise}$$

This signal as well as its auto-correlation function (which is symmetrical if we consider the time period between $-T/2$ and $T/2$) are represented in Figure 5.36 (for a "moderate level" of added noise) and in Figure 5.37 (for a "high" level of added noise).

Regardless of noise, what it can be concluded is that the original time signal has been transformed into a new time signal that tends to zero as $\tau \to \infty$. This means that the Dirichlet condition (Equation 5.61) can now be satisfied, and the Fourier transform can be applied on the auto-correlation function instead. When the Fourier transform pair given earlier by Equations 5.42 and 5.43 is used, the stationary random process is described by the well-known Wiener–Khintchine relationships

$$R_{xx}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{xx}(\omega) e^{i\omega\tau}\,\mathrm{d}\omega \tag{5.64}$$

$$S_{xx}(\omega) = \int_{-\infty}^{+\infty} R_{xx}(\tau) e^{-i\omega\tau}\,\mathrm{d}\tau \tag{5.65}$$

---

[*] The term "ergodic" is used to describe a random process in which the time average of one sample of events is the same as the average of the whole sequence of events.

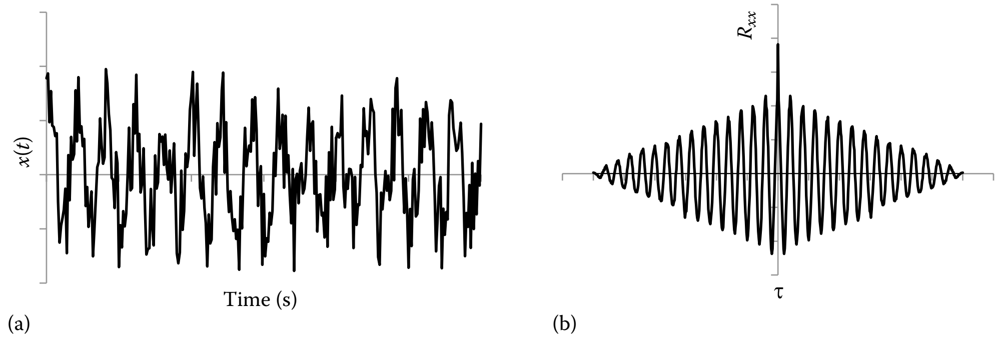


FIGURE 5.36
Example of a harmonic time signal with (a) added random white noise and (b) its auto-correlation function (for a "moderate" level of added noise).

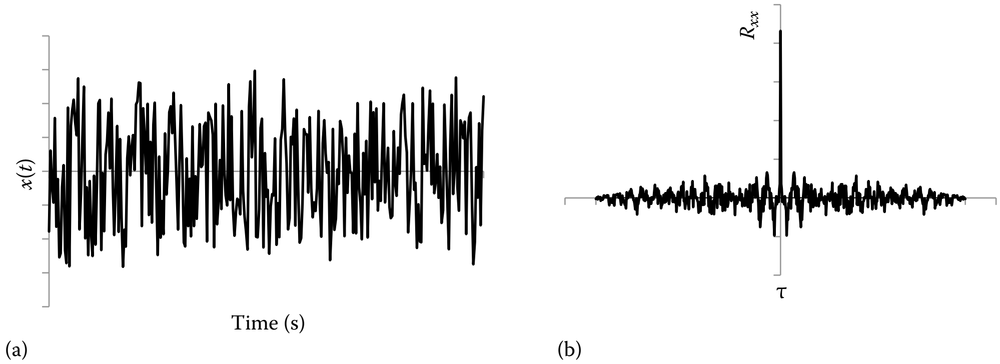


FIGURE 5.37
Example of a harmonic time signal with (a) added random white noise and (b) its auto-correlation function (for a "high" level of added noise).

Equation 5.65 is also known as the auto-spectral density (ASD) or power-spectral density (PSD), which is a real and even symmetric function. The reason why this quantity is called PSD is better understood if we consider $\tau = 0$. In that case, Equation 5.62 combined with Equation 5.64 yields to

$$R_{xx}(0) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x^2(t)\,\mathrm{d}t = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{xx}(\omega)\,\mathrm{d}\omega \tag{5.66}$$

in which $x(t)$ is raised to the power of 2, $x^2(t)$.

The Fourier and power spectra of the time signals shown in Figures 5.36 and 5.37 are represented in Figures 5.38 and 5.39, respectively.

These concepts can further be extended to correlate two different functions, for example, as is the case of the measurement of the so-called FRF. The FRF is a transfer function relating at least one output to one input (for example, the vibration response $x(t)$ to a given

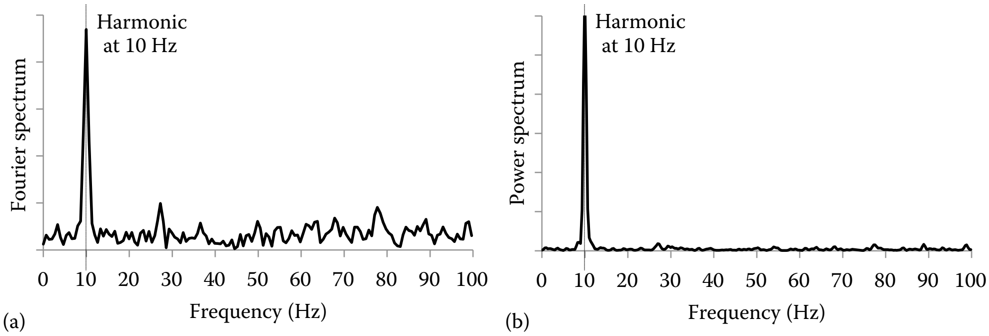


FIGURE 5.38
(a) Fourier spectrum of the time signal shown in Figure 5.36a. (b) Fourier spectrum of the auto-correlation function shown in Figure 5.36b.

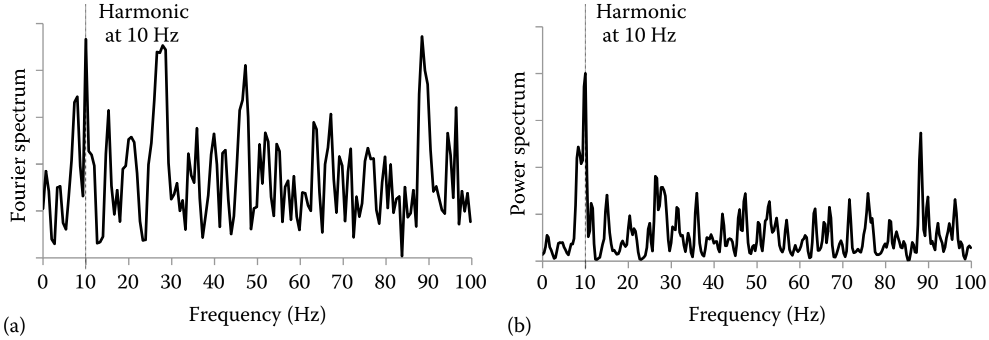


FIGURE 5.39
(a) Fourier spectrum of the time signal shown in Figure 5.37a. (b) Fourier spectrum of the auto-correlation function shown in Figure 5.37b.

random excitation force *f*(*t*)). In this case, the PSD is called cross-spectral density (CSD) or, more often, cross-correlation, and the Wiener–Khintchine relationships become

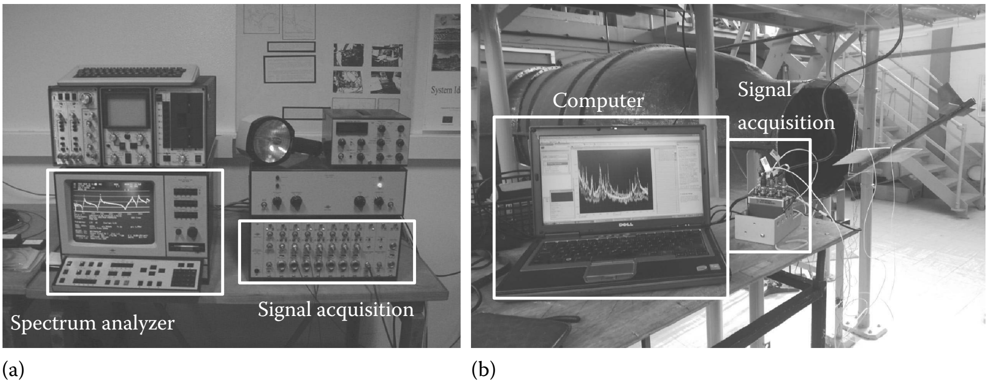
$$R_{fx}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} f(t)x(t+\tau)\,\mathrm{d}t = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{fx}(\omega)e^{i\omega\tau}\,\mathrm{d}\omega \tag{5.67}$$

$$S_{fx}(f) = \int_{-\infty}^{+\infty} R_{fx}(\tau)e^{-i\omega\tau}\,\mathrm{d}\tau \tag{5.68}$$

The CSD functions are complex functions (including real and imaginary parts or magnitude and phase) whereas the PSD is a real function with magnitude only. It can also be shown that [6]

$$\begin{aligned} R_{fx}(\tau) &= R_{xf}(-\tau) \\ S_{fx}(\omega) &= S^{*}_{xf}(\omega) \end{aligned} \tag{5.69}$$

where the superscript * stands for complex conjugate.



(a) (b)

FIGURE 5.40
(a) A spectrum analyzer and signal acquisition unit from Brüel & Kjær. (Photo taken at Instituto Superior Técnico, University of Lisbon, in Portugal.) (b) A Dell laptop computer and National Instruments Co. signal acquisition unit. The photos were taken within a 15-year interval.

It should be noted that the CSD—Equation 5.67—is the convolution between $f(t)$ and $x(t+\tau)$, thus illustrating another application for the convolution earlier described in Section 5.4.4.

Although these concepts are just briefly introduced in this chapter, they are quite important in mechatronics as they are used in a wide range of applications, for example, in condition monitoring, in control, in speech analysis or in electronic communication systems.

In the past, spectral analyzers, although very powerful and reliable, were bulky and expensive equipment. Today, they have been replaced by powerful computers and smaller data acquisition units. Even everyday consumer products, such as mobile phones, are capable of producing spectra from measured time signals (e.g., sounds obtained with the microphone or a mechanical vibration measured with the built-in accelerometer). A comparison between what can be used nowadays and what was the state-of-the-art 20 years ago is shown in Figure 5.40.

5.4.5.2 Estimators

For a harmonic excitation, the FRF is the relationship between the response $x(t)$ (output) to an the excitation $f(t)$ (input):

$$H(\omega) = \frac{x(t)}{f(t)} \tag{5.70}$$

which, in terms of block diagram, can be represented as in Figure 5.41.

The function $H(\omega)$ is a transfer function that represents the system. However, it is very important to note that Equation 5.70 is not valid unless the signals are harmonic,* thus the importance of the auto-correlation and further developments previously explained for nonperiodic signals.

---

* It is also important to note that many authors do not like the notation used in Equation 5.70. This is because we are equating a function that depends on frequency with a function that depends on time. However, some textbooks [6] use it for the sake of better understanding of the FRF.

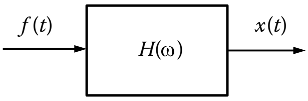


FIGURE 5.41
Frequency response function (FRF) for a harmonic excitation on a single-input-single-output (SISO) system.

The convolution of the response with force can, however, be applied. It is possible to show that [15]

$$|H(\omega)|^2 = \frac{S_{ff}(\omega)}{S_{xx}(\omega)} \tag{5.71}$$

and

$$H_1(\omega) = \frac{S_{xf}(\omega)}{S_{xx}(\omega)} \tag{5.72}$$

Equation 5.72 represents one of the three versions for the FRF estimators, and that is why an index of 1 was added next to the letter H. This estimator is the "conventional" FRF estimator and is determined by using the cross input–output spectrum and the input auto-spectrum. It is said to be unbiased with respect to noise on the output.

Another version of the FRF estimator, $H_2(\omega)$, is obtained by dividing the output auto-spectrum by the cross input–output spectrum:

$$H_2(\omega) = \frac{S_{ff}(\omega)}{S_{fx}(\omega)} \tag{5.73}$$

This estimator is said to be unbiased with respect to noise on the input.

In principle, both estimators $H_1(\omega)$ and $H_2(\omega)$ should yield the same result. However, in practice, this may not happen, due to many reasons:

1. The signals contain noise.
2. The system relating the input and output is not linear.
3. The measured output $x(t)$ is not a consequence of the input $f(t)$ alone, but also from other nonquantified external inputs.

Hence, an indicator of the quality of the analysis can be defined as the ratio of the two estimators:

$$\gamma^2(\omega) = \frac{H_1(\omega)}{H_2(\omega)} = \frac{|S_{xf}(\omega)|^2}{S_{xx}(\omega)S_{ff}(\omega)} \tag{5.74}$$

where $\gamma^2(\omega)$ is called the *ordinary coherence function*, which can assume values in the interval [0,1]. Basically, the coherence can be seen as a measure of the "correlation" between two signals. Here, the term "correlation" does not have the sense of statistical correlation, but linear relation instead.

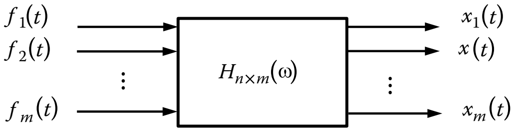


FIGURE 5.42
Frequency response function (FRF) for a harmonic excitation on a multiple-input-multiple-output (MIMO) system.

For MIMO systems (Figure 5.42) with m inputs written in the vector form $\{f(t)\}$ and n outputs written in the vector form $\{x(t)\}$, this can be further generalized to obtain, for example,

$$[H(\omega)]^T = [S_{xx}(\omega)]^{-1}[S_{fx}(\omega)] \tag{5.75}$$

assuming $[S_{xx}(\omega)]$ is invertible and $[H(\omega)]$ is an $n \times m$ matrix of frequency response functions.

5.4.5.3 Ensemble Averaging

As mentioned before, the auto-correlation is a function that is used to find patterns in random events. Nevertheless, signals in random processes are inherently noisy by nature. If noise is random, one way to cancel it out (or, at least, reduce it) is by using *ensemble averaging.*

Typically, spectrum analyzers bring two ensemble averaging weighting options: linear or exponential. According to the LabVIEW™ help documentation [16] on spectral measurements, linear and exponential averaging operate in the following way:

1. Linear: Averages are made over a specified number n of measurement time periods T in a nonweighted manner, that is, each set of data weighs exactly the same as in an arithmetic mean.
2. Exponential: Averages are made over a specified number n of measurement time periods T in a weighted manner. Exponential averaging gives the most recent sets of data more weighting in the average than older data.

Averaging is performed on spectral quantities. Thus, the following options are generally available as well [16]:

1. Vector averaging: The average of complex FFT spectra quantities is computed directly. Vector averaging eliminates noise from synchronous signals.
2. RMS averaging: Averages the energy, or power, of the FFT spectrum of the signal.
3. Peak hold: Performs averaging at each frequency line separately, retaining peak levels from one FFT record to the next.

Another method to reduce noise consists of using *overlapping* in what is also known as Welch's method. Overlapping is often defined in terms of a percentage of the time signal. The time signal is split into a set of segments that contains shifted data with finite duration. These segments are then overlapped by a percentage of the total number of points in one segment. When overlapping is 0%, this process is known as Barlett's method. Overlapping segments are then windowed. Because most window functions give more importance to the data at the center of the set of data, the overlapping technique compensates for that.

5.4.6 Butterworth Filter

Filters are used to remove unwanted contents from a signal. For example, there are filters that can selectively remove certain features, such as noise or spurious spectral components, to avoid error propagation within an algorithm. Another example might be to simulate how another sensor rather than the one being used would perceive a signal, for example, filtering a sound obtained with a microphone to represent how the human hearing would have perceived it.

Filters can be linear, nonlinear, time-invariant, analog, digital, passive, active, etc. Filters can also be low-pass, high-pass, band-pass, or band-reject. Low-pass filters are designed to cut off high frequencies whereas high-pass filters are the exact opposite. Band-pass filters are designed to pass all the signal components within a finite frequency band, blocking away the components outside that band. Band-reject is the opposite to band-pass.

One particular example that most of us are familiar with is the Dolby noise reduction system used in audio. Basically, the Dolby system consists on the use of band-pass filters, both during recording and playback. In essence, the Dolby system consists of an encode–decode system in which the amplitude frequencies in one band are increased during recording and decreased during playback, proportionally. Basically, what this does is to increase the signal-to-noise ratio (SNR) rather than filtering out noise as in subsequent DSP.

Linear continuous time filters are among the most used filters in signal processing, out of which the low-pass Butterworth filter probably is the most popular. Nevertheless, filters like the Chebyshev (types I and II), elliptical, and others are quite popular as well. The three aforementioned are time-invariant filters that are based on their analog counterparts and fit in the IIR (infinite impulse response) category. Filters can also be FIR (finite impulse response), depending on the way they are implemented. IIR filters are the most efficient to implement in DSP. However, IIR filters are not as stable as FIR filters are because phase response is not linear, and IIR is difficult to implement in hardware while FIR can be efficiently realized on hardware. Ultimately, IIR filters are less accurate than FIR but faster in a way.

The Butterworth filter is a low-pass filter that is designed to block off all signal components above a certain frequency limit while keeping the frequency response's passband as flat as possible (with uniform sensitivity). The magnitude of its transfer function (frequency response function) is represented in Figure 5.43 and given by

$$\left|H(i\omega)\right| = \frac{1}{\sqrt{1+\left(\dfrac{i\omega}{i\omega_c}\right)^{2N}}} \tag{5.76}$$

where N is the order (number of poles) in the filter and ω_c is the cutoff frequency in *rad/s*.

The Butterworth filter has a slow roll-off (decay) when compared to other filters such as the Chebishev or elliptic (equiripple) filters although it does not present ripple (a small residual periodic variation). When the graphs in Figure 5.43 are presented in a log–log plot (Figure 5.44), it is possible to have a better understanding of how the filter rolls off with frequency for different numbers of poles. For example, in a second-order Butterworth filter, when the frequency increases by a factor of 10 (i.e., one decade), the magnitude drops 40 dB. In other words, the roll-off is −40 dB/decade.

The order of the filter is quite important as it defines how the filter rolls off in the frequency domain. The higher the number of poles, the faster it rolls off and the higher the attenuation in the stopband. However, this will, of course, lead to more complicated algorithms and circuits as Butterworth filters are based on polynomials.

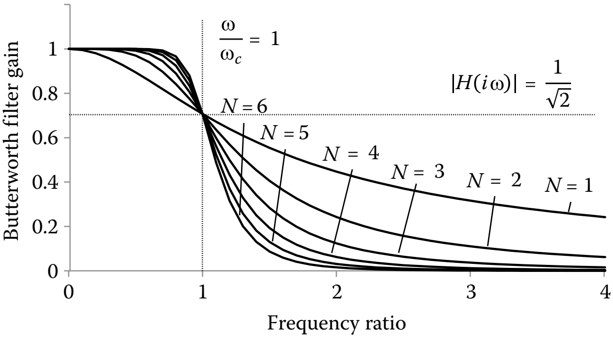


FIGURE 5.43
Butterworth low-pass filter gain amplitude for a different number of poles.

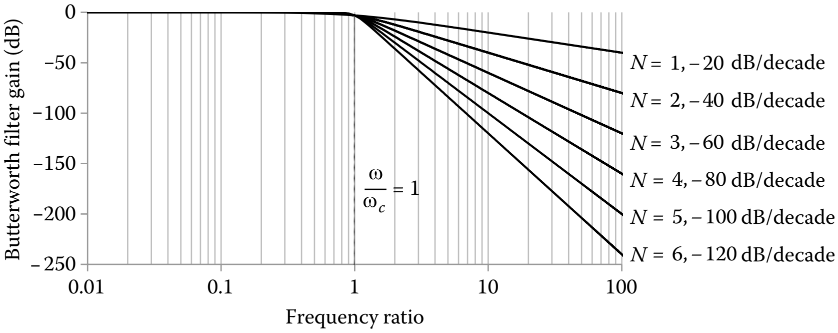


FIGURE 5.44
Butterworth low-pass filter gain amplitude and decay for a different number of poles (log–log plot).

The design of a digital filter requires mathematical manipulation that is outside the scope of this text. In the case of the Butterworth low-pass filter, the steps can generically be summarized into the following:

1. Define the filter topology (cutoff frequency, ripple, out of band attenuation, etc.).
2. Transform the continuous time domain filter into a discrete time domain filter. This can be done, for example, using a bilinear transform.
3. Transform the frequency response function from the frequency domain to the time domain. This can be done by using the z-transform. The z-transform is a special case of the Fourier transform in which $e^{i\omega}$ is replaced by the variable $z = (e^{i\omega})$.

Generalizing for any even-order Butterworth low-pass filter, the discrete-time system transfer function can be expressed as [17]

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-k}}{1 + \sum_{k=1}^{N} b_k z^{-k}} \tag{5.77}$$

where

$$
\begin{aligned}
a_0 = a_k &= \frac{\Omega_c^2}{c(k)}, k = 0, 2, 4, 6, \ldots (k \text{ even}) \\
a_k &= 2a_{k-1}, k = 1, 3, 5, \ldots (k \text{ odd}) \\
b_k &= \frac{2}{c_k} - 1, k = 2, 4, 6, \ldots (k \text{ even}) \\
b_k &= \frac{2\left(\Omega_c^2 - 1\right)}{c_k}, k = 1, 3, 5, \ldots (k \text{ odd})
\end{aligned}
\tag{5.78}
$$

and

$$
\begin{aligned}
c_k &= 1 + 2\cos\left[\frac{\pi(2k+1)}{2N}\right]\Omega_c + \Omega_c^2 \\
\Omega_c &= \tan\left(\pi\frac{f_c}{f_s}\right) \\
N &= 0, 2, 4, 6, \ldots
\end{aligned}
\tag{5.79}
$$

where f_c is the cutoff frequency, and f_s is the sampling frequency, both in Hz. The filtered time signal is [17]

$$
x'(t) = \sum_{k=0}^{N} a_k x(t-k) - \sum_{k=1}^{N} b_k x'(t-k) \tag{5.80}
$$

The Butterworth filter is a recursive filter: It requires previous filtered time samples (as much as the number of poles N). Because the first filtered time samples are not available, initial guessing is required. One way of doing this consists of, for example, padding (adding as many zeros at the beginning of the signal as the unknown data) or assuming that the first samples of the filtered signal are the same as the unfiltered one.

Let us take as an example the following waveform:

$$
x(t) = \sin(14\pi + \pi/3) + \text{random noise}
$$

This is a 7-Hz sine wave with added noise. Let us assume this signal was acquired with a sampling frequency of 100 Hz and that we want to set the digital cutoff frequency at 10 Hz. Also, consider a 40 dB roll-off, that is, a second-order filter. In this case, the Butterworth polynomial coefficients are

$$
\begin{aligned}
a_0 &= 0.067455 \\
a_1 &= 0.134911 \\
a_2 &= 0.067455 \\
b_1 &= -1.14298 \\
b_2 &= 0.412801
\end{aligned}
$$

Figures 5.45 and 5.46 show the result of the use of this second-order Butterworth filter on the time and frequency domains, respectively. The Butterworth filter was very effective in minimizing high-frequency noise (practically eliminated it completely) although it was not as effective in the 10- to 20-Hz frequency range. On the other hand, due to the slow and progressive roll-off of the Butterworth filter, the amplitude of the component at 7 Hz was slightly attenuated by the filter even if it was below the 10-Hz cutoff frequency.

The op-amp circuit for the Butterworth second-order low-pass filter can take the Sallen–Key topology shown in Figure 5.47 [18,19].

Defining the constants:

$$\begin{aligned}\tau_1 &= R_1C_1\\ \tau_2 &= R_2C_2 \qquad (5.81)\\ \tau_3 &= R_1C_2\end{aligned}$$

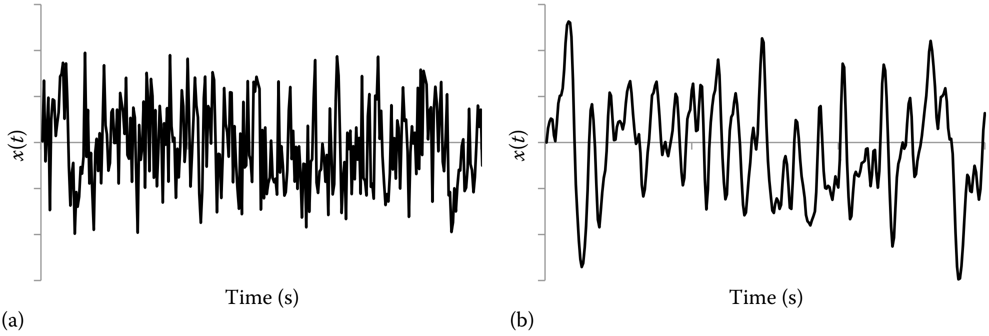


FIGURE 5.45
(a) Time waveform of a 7-Hz sine wave with added random noise. (b) The same time waveform after a Butterworth filter is applied with a cutoff frequency at 10 Hz.

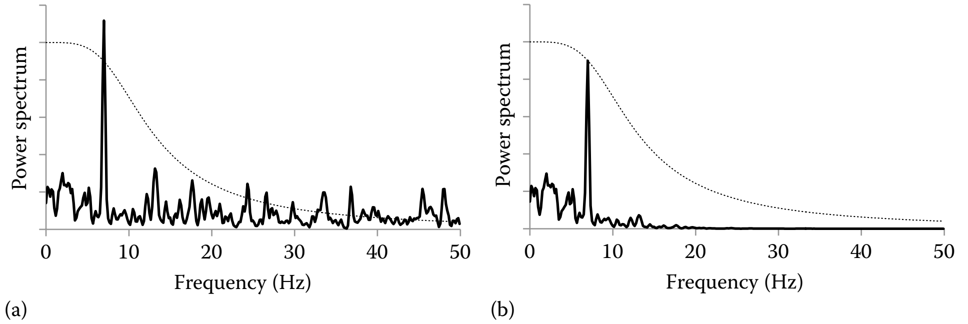


FIGURE 5.46
(a) Power spectrum of the time signal shown in Figure 5.45a. (b) Power spectrum of the time signal shown in Figure 5.45b (after being filtered with a low-pass Butterworth filter). The dashed line in both pictures represents the Butterworth frequency response function.

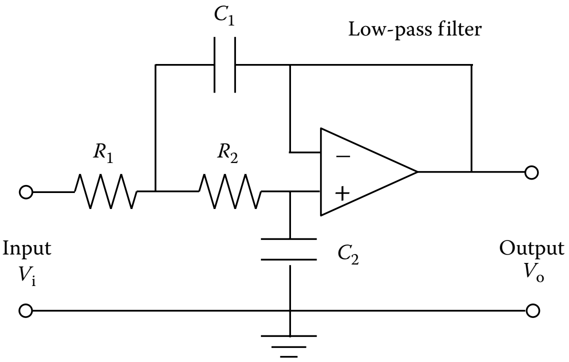


FIGURE 5.47
Op-amp circuit of a Butterworth second-order low-pass filter.

the transfer function for the circuit shown in Figure 5.47 is [18]

$$H(s) = \frac{V_o}{V_i} = \frac{1}{\tau_1\tau_2 s^2 + (\tau_2 + \tau_3)s + 1} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.82}$$

where s is the Laplace variable, ω_n is the undamped natural frequency, and ζ is the damping ratio. From Equation 5.82, the undamped natural frequency and the damping ratio are, respectively,

$$\omega_n = \frac{1}{\sqrt{\tau_1\tau_2}} \tag{5.83}$$

$$\zeta = \frac{\tau_2 + \tau_3}{2\sqrt{\tau_1\tau_2}} \tag{5.84}$$

For this second-order transfer function to become oscillatory, is necessary that $\zeta < 1$. Furthermore, ideally, the system should have a zero-resonant frequency $\omega_r = 0$:

$$\omega_r = \sqrt{1 - 2\zeta^2}\,\omega_n = 0 \tag{5.85}$$

Thus,

$$\zeta = \frac{1}{\sqrt{2}} \tag{5.86}$$

and, as a result,

$$(\tau_2 + \tau_3)^2 = 2\tau_1\tau_2 \tag{5.87}$$

The zero-resonant frequency ω_r must be zero so that the gain is flat and unitary until the cutoff frequency. If, instead of the transfer function, one plots the amplification factor (also

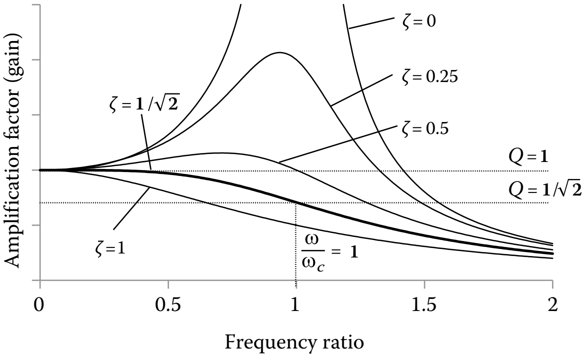


FIGURE 5.48
Plots of the amplification factor for different values of the damping ratio.

called gain), this might be easier to understand (Figure 5.48). The amplification factor of any second-order oscillatory system (either mechanical or electrical) is given by [6]

$$Q(\beta) = \frac{1}{\sqrt{(1-\beta^2)^2 + (2\zeta\beta^2)}} \tag{5.88}$$

where β is the frequency ratio:

$$\beta = \frac{\omega}{\omega_n} \tag{5.89}$$

What Figure 5.48 shows us is that for a value of the damping ratio $\zeta = 1/\sqrt{2}$, the maximum amplification factor (gain) is 1. At a value of the frequency ratio β = 1, the amplification factor is $Q = 1/\sqrt{2}$. This is equivalent to what is illustrated in Figure 5.43: For a unitary frequency ratio, the filter's gain is $1/\sqrt{2}$. Thus, what this tells us is that ω_n in Equation 5.89 actually is the cutoff frequency ω_c, that is,

$$\omega_c = \frac{1}{\sqrt{\tau_1\tau_2}} = \frac{1}{\sqrt{R_1R_2C_1C_2}} \tag{5.90}$$

In his original work [20], Butterworth also showed that it is possible to design a high-pass filter from the low-pass counterpart by switching the inductances with the capacitances and vice versa. Furthermore, he expanded the concept to band-pass filters. The corresponding op-amp circuits forsecond-order filters are illustrated in Figure 5.49.

5.4.7 Smoothing Filters

Smoothing is a technique that is used to reduce noise across the whole frequency range of a signal without filtering out meaningful components. Typically, smoothing consists of constructing an approximating function that attempts to capture important patterns in a signal. However, this must not be confused with curve-fitting methods as these involve

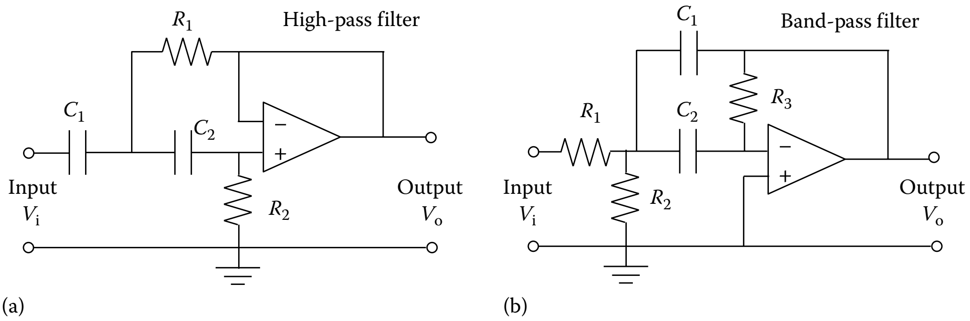


FIGURE 5.49
Op-amp circuits of Butterworth second-order (a) high-pass and (b) band-pass filters.

the use of an explicit function form for the result (e.g., with physical meaning) whereas smoothing techniques, strictly speaking, do not.

5.4.7.1 Moving Average

Many smoothing algorithms have been developed for signal processing. One of the most popular is the *moving average*, with which a series of averages of different subsets of the data set are created. This should not be confused with the ensemble averaging earlier introduced. Although ensemble averaging operates in successive sets of data, the moving average operates on a single data set exclusively. Ensemble averaging is in itself a particular technique of smoothing although it is not a filter.

The moving average works by replacing each data point with an averaged value of the neighboring data points defined within the span. This is similar to a low-pass filter (see Section 5.4.6) with the smoothing function being given by [21]

$$x_s'(t) = \frac{1}{2N+1}\sum_{i=-N}^{N} x(t+i) \tag{5.91}$$

where $x_s'(t)$ is the smoothed value for the tth data point, N is the number of neighboring data points on either side of $x_s'(t)$, and $2N + 1$ is the span.

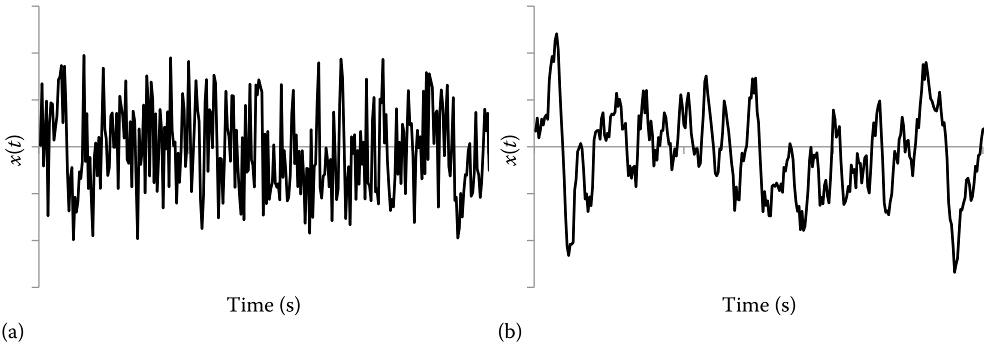


FIGURE 5.50
(a) Time waveform of a 7-Hz sine wave with added random noise. (b) The same time waveform after a moving average smoothing filter is applied with $N = 3$.

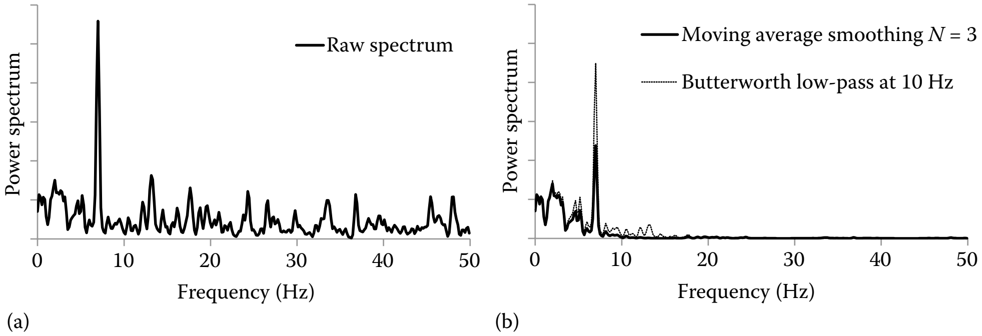


FIGURE 5.51
(a) Power spectrum of the time signal shown in Figure 5.50a. (b) Power spectra of the time signals shown in Figures 5.50b (after being smoothed with a moving average with $N = 3$) and 5.45b (filtered with a low-pass Butterworth filter with cutoff frequency at 10 Hz).

As an example, the waveform originally presented in Figure 5.45a and to which a Butterworth low-pass filter was applied, resulting in Figure 5.45b, has now been *smoothed* using the moving average Equation 5.91. The resulting smoothed waveform, for a value of $N = 3$, is shown in Figure 5.50, with corresponding power spectra shown in Figure 5.51.

As with the Butterworth filter, it can be observed that high-frequency noise has been practically eliminated although it still is visible in the low-frequency range. Nevertheless, the amplitude of the existing harmonic has been highly attenuated as a result even more than when the Butterworth filter was used.

5.4.7.2 Savitzky–Golay

The Savitzky–Golay filtering is another popular technique used for smoothing. It can be seen as a generalized weighted moving average. The filter has coefficients that are derived by fitting a polynomial of a chosen degree with least squares fit. The advantage of the Savitzky–Golay filtering against the simple moving average is that it allows reducing attenuation of data features by increasing the order of the polynomials.

Savitzky and Golay [22] showed that a set of integers c_i could be derived and used as weighting coefficients to carry out the smoothing operation. The use of these weighting coefficients, known as convolution integers, turns out to be exactly equivalent to fitting the data to a polynomial and is computationally more effective and much faster [23]. The smoothed data point $x_s'(t)$ by the Savitzky–Golay algorithm is given by the following equation:

$$x_s'(t) = \frac{\sum_{i=-N}^{N} c_i x(t+i)}{\sum_{i=-N}^{N} c_i} \tag{5.92}$$

where coefficients c_i are obtained through a least squares fit, and $c_i = c_{-i}$. Typical coefficients for quadratic and cubic Savitzky–Golay filters are shown in Table 5.6. Usually,

TABLE 5.6

Typical Coefficients for Quadratic and Cubic Savitzky–Golay Filters

N	$2N+1$	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
2	5	17	12	–3										
3	7	7	6	3	–2									
4	9	59	54	39	14	–21								
5	11	89	84	69	44	9	–36							
6	13	25	24	21	16	9	0	–1						
7	15	167	162	147	122	87	42	–13	–78					
8	17	43	42	39	34	27	18	7	–6	–21				
9	19	269	264	249	224	189	144	89	24	–51	–136			
10	21	329	324	309	284	249	204	149	84	9	–76	–171		
11	23	79	78	75	70	63	54	43	30	15	–2	–21	–42	
12	25	467	462	447	422	387	343	287	222	147	62	–33	–138	–253

Note: $c_i = c_{-i}$.

Savitzky–Golay filters are applied to odd-sized windows ($2N + 1$) only although the concept has further been extended for even-sized windows ($2N$) more recently [24].

A quadratic/cubic Savitzky–Golay smoothing filter with $N = 3$ was used in the same example as earlier presented in both Sections 5.4.6 and 5.4.7, resulting in Figures 5.52 and 5.53.

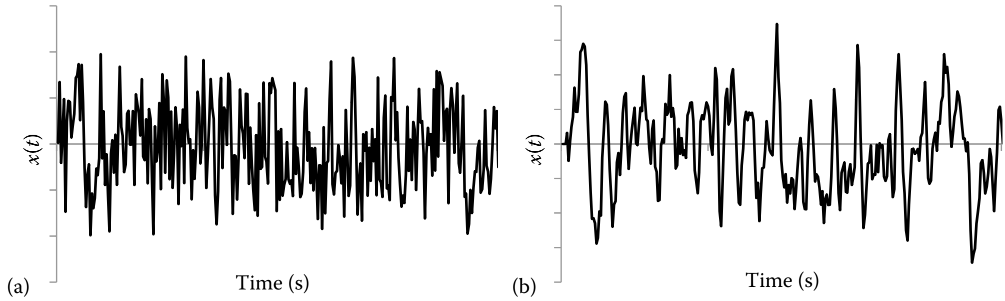


FIGURE 5.52
(a) Time waveform of a 7-Hz sine wave with added random noise. (b) The same time waveform after a Savitzky–Golay smoothing filter is applied with $N = 3$.

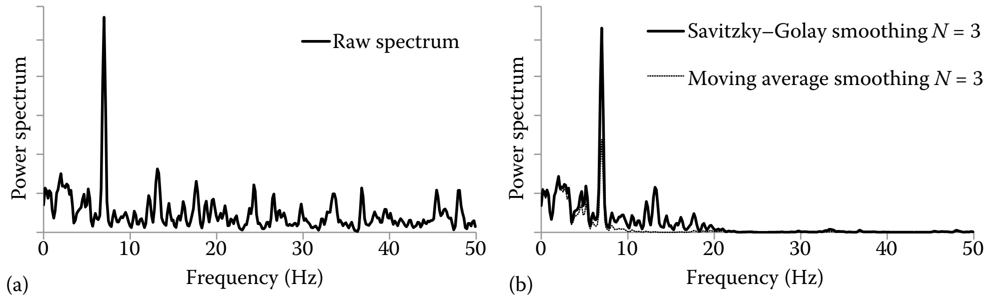


FIGURE 5.53
(a) Power spectrum of the time signal shown in Figure 5.52a. (b) Power spectra of the time signals shown in Figures 5.52b (after being smoothed with a Savitzky–Golay filter) and 5.50b (after being smoothed with a moving average with $N = 3$).

In comparison to the moving average smoothing (and even to the Butterworth filter), it is clear that the Savitzky–Golay smoothing filter preserved much better the spectral component at 7 Hz (which is a data feature) even if it was not as effective in noise attenuation.

5.5 Sensors

Although there may exist other types of sensor classifications, most sensors would fit into active or passive and analog or digital. Active sensors require external power for their operation. An example is an integrated electronic piezoelectric (IEPE) accelerometer. On the contrary, a passive sensor does not need external power for operation as is the case of a charge accelerometer.

An analog sensor produces an output that is proportional to the variable being measured. The relationship between the output and the variable being measured is not necessarily linear although typically it is within the operating range. This relationship is called the calibration factor (or curve). Typically, the maximum analog output voltage is 10 V, and the maximum analog current is between 4 and 20 mA.

A digital sensor is one that converts the measuring signal into a digital signal before it is transmitted to the acquisition system. Typically, it has built-in electronics. As an advantage when compared to the analog sensor, the digital data transmission is not sensitive to cable length, resistance, or impedance and is not affected by electromagnetic noise.

A schematic of a generic sensor setup is shown in Figure 5.54.

In this section, an introductory overview of conventional sensors used in mechatronics and applications is presented. However, in this book, we do not have the intent to deliver an all-encompassing review of sensor technology and fundamentals. It must be noted that sensors are a vast area that already deserved the publication of entire textbooks on the topic (e.g., Ref. [18]).

5.5.1 Accelerometers

Accelerometers measure motion in the form of acceleration. When used in vibration applications, typical units are ms^{-2} or g(1 g = 9.81 ms^{-2}).

5.5.1.1 Piezoelectric Accelerometers

Piezoelectric accelerometers are composed of three main components: a base with a central post, a piezoelectric crystal, and a seismic mass (Figure 5.55). The base is attached to the structure via a stud (thread), glue, an adhesive pad, bee's wax, or a magnetic base.

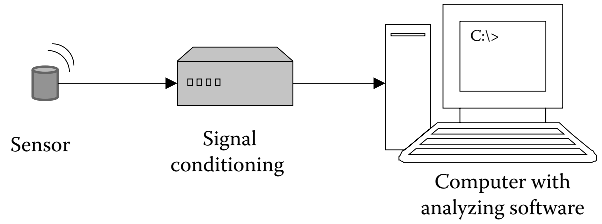


FIGURE 5.54
Generic sensor setup.

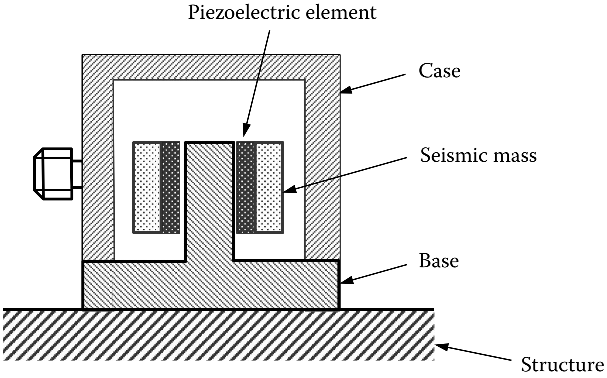


FIGURE 5.55
Schematic cross-sectional view of a piezoelectric accelerometer.

The base of the accelerometer moves with the motion of the structure, and because the piezoelectric element and seismic mass are attached to the central post, they will follow motion. It must be noted that the piezoelectric element has some elasticity. Thus, the seismic mass and the piezoelectric element behave as a mass–spring single-degree-of-freedom (SDOF) system to which motion is transmitted. When the piezoelectric crystal is deformed, a charge in pC (pico-Coloumb) is produced. This charge is proportional to acceleration, according to Newton's second law.

These devices typically work in a fairly wide range of frequencies, typically between 1 Hz and 30 kHz. Because the seismic mass and piezoelectric element constitute the elements of a SDOF, the accelerometer itself has a resonant frequency that is usually shown in the calibration chart. A rule of thumb is that an accelerometer is usable up to one third its resonant frequency. In the example shown in Figure 5.56 (Brüel & Kjær charge accelerometer type 4501) the resonant frequency of the accelerometer is 50 kHz, which means that, according to this rule of thumb, the accelerometer is usable (with approximate constant

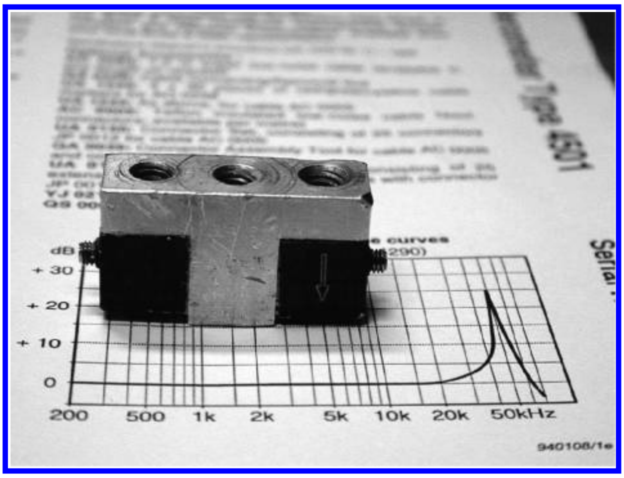


FIGURE 5.56
Calibration chart of a Brüel & Kjær charge accelerometer type 4501 showing its frequency curve. The two accelerometers are mounted on a T-shaped block used to measure rotations. The accelerometers are seen as the two black regions in the bottom corners of the unit. The measurement axis is the vertical axis of the accelerometer.

sensitivity) up to 16.7 kHz. This value actually is very close to the one presented by the manufacturer in its calibration chart (16.6 kHz) [25].

The charge accelerometer fits in the passive analog sensor category. Before the signal is sent to the acquisition unit or computer, it is necessary that the charge in pC is converted to voltage in mV (millivolt) in a device called the charge amplifier. The setup of a charge accelerometer is shown in Figure 5.57. As a limitation, it requires the use of an expensive low-noise cable between the accelerometer and charge amplifier, and the measurement is sensitive to the low-noise cable's length. On the other hand, as an advantage, charge amplifiers such as the one shown in Figure 5.25 allow adjusting the dynamic range, which is made by setting the output value in terms of mV/ms^{-2}.

A more modern type of accelerometer is the IEPE, which has got a built-in pre-amplifier. In this case, the charge amplifier is not needed, but the acquisition unit must power the accelerometer (Figure 5.58). It is also recommended that the acquisition unit has an anti-aliasing low-pass filter before the ADC is performed.

In comparison to the charge accelerometer, the IEPE has the advantage of having a fixed sensitivity regardless of cable length and quality, being more portable (practically plug-and-play), and requiring less expensive conditioners. Nevertheless, as limitations, it has a smaller (and fixed) dynamic range, and its operating temperature is limited (maximum of 120°C against up to 480°C for the charge accelerometer). Furthermore, if the electronics in the IEPE are damaged due to, for example, mishandling or dropping on a hard floor, the accelerometer will not work whereas a charge accelerometer may still be functional even if the sensitivity may have changed.

Piezoelectric accelerometers are used in many applications from condition monitoring to structural health monitoring (SHM) or modal analysis. An example of an application is illustrated in Figure 5.59, in which the vibration levels on a generator's eight-cylinder engine are being measured at different coordinates and directions.

Typically, accelerometers measure linear acceleration in a single direction (although they still have a small transverse sensitivity up to 4%). There also exist triaxial accelerometers that can measure along the three directional axes, but they are more bulky and expensive.

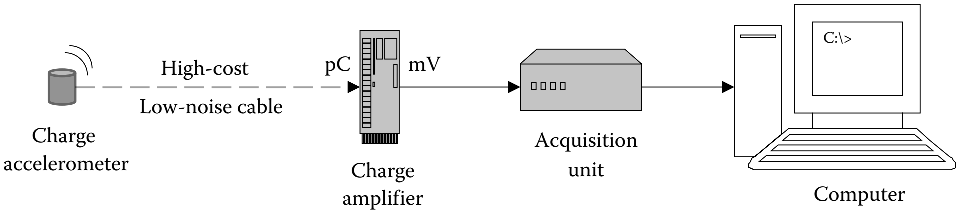


FIGURE 5.57
Charge accelerometer setup.

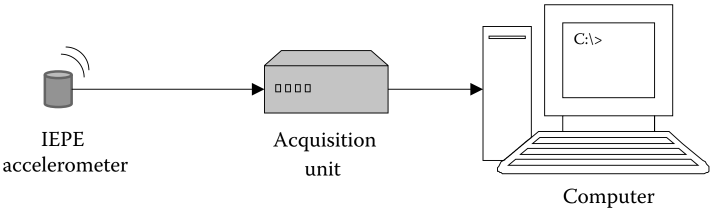


FIGURE 5.58
IEPE accelerometer setup.

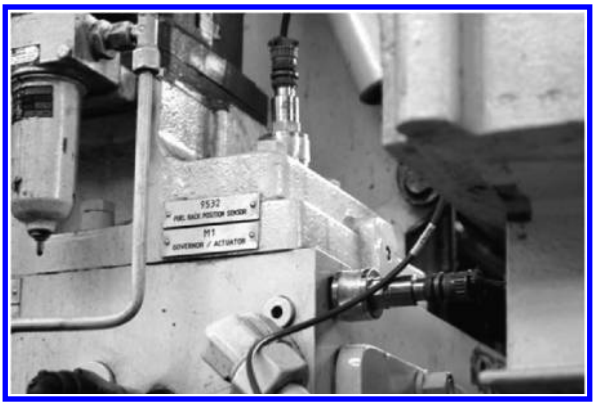


FIGURE 5.59
Two IEPE accelerometers from CSI are being used to monitor the vibration level on a generator's eight-cylinder engine at different coordinates and directions. The accelerometers are attached to the structure by means of two threaded magnets.

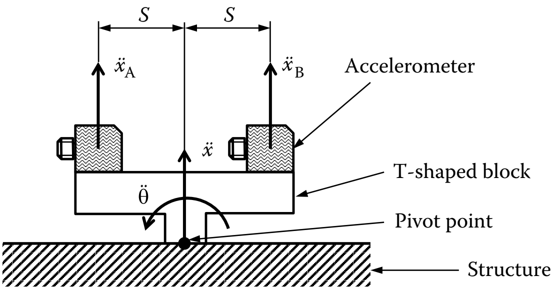


FIGURE 5.60
Schematics of the T-shaped block technique to measure rotations with linear (translational) accelerometers.

For the measurement of structural rotations, piezoelectric accelerometers have not proven as efficient. In a technique that is many times referenced in the literature [6,26,27], two piezoelectric accelerometers are attached at the tips of a solid T-shaped block (usually made from aluminum) as represented in Figure 5.60 (a photo of a T-shaped block was previously shown in Figure 5.56).

Once the accelerations $\ddot{x}_A$ and $\ddot{x}_B$ are known and as long as the frequency range is small enough to keep the T-shaped block behaving as a rigid body even if the structure is not, the linear and angular accelerations at the pivot point *O* of the structure can be determined, with reasonable accuracy, from

$$\ddot{x} = \frac{\ddot{x}_A + \ddot{x}_B}{2} \tag{5.93}$$

$$\ddot{\theta} = \frac{\ddot{x}_A - \ddot{x}_B}{2s} \tag{5.94}$$

5.5.1.2 Piezoresistive and Capacitive Accelerometers

The piezoresistive and capacitive accelerometers work from the principle of deformation of a cantilever-like beam (Figure 5.61). When the structure moves, the seismic mass will produce

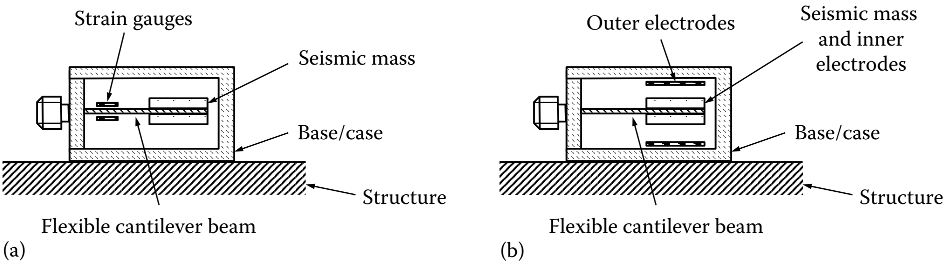


FIGURE 5.61
Schematic cross-sectional views of (a) a piezoresistive accelerometer and (b) a capacitive accelerometer.

a force proportional to acceleration (Newton's second law) deflecting the flexible cantilever beam. In the piezoresistive accelerometer, strain gauges are used to measure strain in the flexural element (strain gauges are later discussed in Section 5.5.4). The capacitive accelerometer is based on the principle that, as the seismic mass moves from its equilibrium position, the capacitance between the electrodes is changed proportionally to acceleration.

The greatest advantage of the use of piezoresistive or capacitive accelerometers (when compared to piezoelectric accelerometers) is that they allow DC measurements (0 Hz). As limitations, they are very sensitive to temperature changes, and the dynamic range is limited due to their higher low linearity.

5.5.2 Velocity Transducers

5.5.2.1 Laser Doppler Velocimeters

Laser Doppler velocimeters (LDVs) measure velocity of a moving target using the Doppler effect. When used for vibration measurements, typical units are mms^{-1} or ips.

"Laser" is the English acronym for *light amplification by stimulated emission of radiation*. Its functioning is based on the stimulated emission of photons: Light is collimated. Two waves are said to be collimated if they have the same length, a constant relative phase, and travel at parallel directions (consequently, collimated light is monochromatic as well).

He–Ne laser vibrometers produce red light in the visible spectrum with a wavelength of 632.8 nm. The principle behind a laser vibrometer is interference, that is, the addition between two collimated waveforms. When two laser beams interfere with one another, there is a phase and wavelength relationship between the two waves that can be determined even if the beams have traveled different distances. In other words, the relationship between the light that is emitted to a target and the light that is reflected from that target contain information about the velocity and position of the moving object. It can be said that motion modulates the phase of the light wave whereas velocity modulates the optical frequency. However, because a He–Ne laser light has a frequency of 4.74×10^{14} Hz, it is not practical (or possible) to demodulate the signal directly, and interferometry techniques must be used [28].

In the modified Mach–Zehnder interferometer (Figure 5.62), the reflected light is combined with a reference beam so that both the emitted and reflected signals interfere with one another. Inside the interferometer, the laser is split into reference and test beams in a polarizing beam splitter PBS1. The reference beam f_0 is transmitted to the photodiode PD passing through a mirror M, Bragg cell BC, and beam splitter BS2. The Bragg cell is a device that adds a known frequency f_B shift to the reference beam f_0, that is, $f_0 + f_B$. This

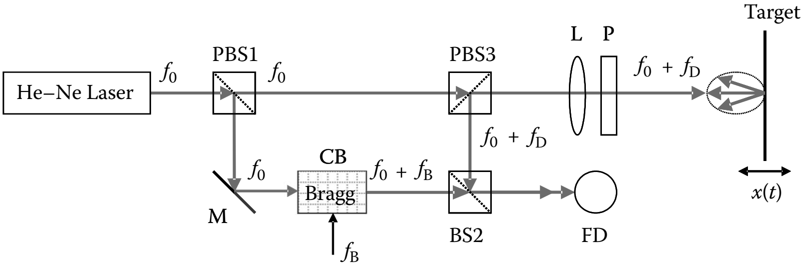


FIGURE 5.62
Schematics of the principle of operation of a Mach–Zehnder laser interferometer.

frequency f_B is called the carrier frequency, usually generated by quartz at 40 MHz. The test beam f_0 is sent directly to the target in motion and reflected back. The motion of the target adds a Doppler shift f_D to the reflected light, so it becomes $f_0 + f_D$. The reflected light comes in through the system of lenses P and L and beam splitter BS2, reaching the photodiode PD where the reference and test beams are collected and interfered.

The use of a laser vibrometer has some advantages when compared to the use of accelerometers. One of the greatest advantages is that it is a contactless technology, meaning it does not add mass to the system. The effects of added mass may be relevant in vibration measurements, especially for lightweight systems. Moreover, being contactless means it can measure moving parts such as rotating shafts or turbine blades, it can measure locations of otherwise difficult access, and it can be used to measure high-temperature surfaces. In terms of disadvantages, it requires that the test surfaces are somewhat reflective (reflectors can be used), it is a considerably expensive technology, and, as with any nonseismic sensor, it may be affected by its referential, for example, external ground motion.

In terms of advanced applications, LDV suppliers offer scanning and 3-D measurement solutions although the cost may become a hindrance to the use of such technology. Figure 5.63 shows an experimental setup in which a scanning laser vibrometer is used for

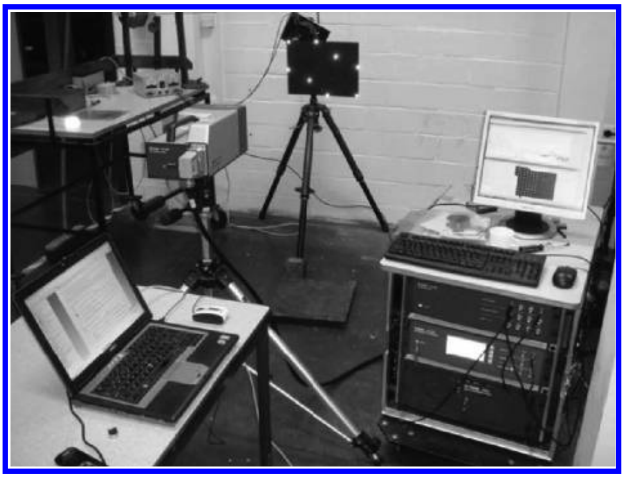


FIGURE 5.63
Test setup for the measurement of the vibration modes of a free–free suspended carbon fiber plate with a Polytec scanning LDV. (From Montalvão, D., *A Modal-Based Contribution to the Damage Location in Laminated Composite Plates*, PhD Thesis, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, 2010. Photo taken at the Vrije Universiteit Brussel in Belgium.)

the development of a SHM technique on composite materials based on the assessment of mode shape changes [29,30].

5.5.2.2 Tachometers

A tachometer is a device that is used to measure velocity on a rotating object, such as a shaft, disk, or toothed gear. Typical units are rpm (rotations per minute) although this can be easily converted to other more convenient units, such as cps or Hz (cycles per second) or the SI unit rad/s:

$$1 \text{ rpm} = \frac{1}{60}\text{Hz} = \frac{2\pi}{60}\text{rad/s} \tag{5.95}$$

One example of a tachometer is shown in Figure 5.64. This is a handheld digital tachometer that measures the angular velocity of the rotating disk probe at the tip. The probe's surface is coated with rubber to increase friction. With this particular instrument, it is possible to either measure the rotation on a disk or shaft or to measure the linear velocity on straight objects such as the conveyor belt shown.

One type of sensor that is frequently used to measure velocity on rotating shafts is the proximity probe. The main difference between the proximity probe and the tachometer shown in Figure 5.64 is that the proximity probe is a contactless device. Proximity

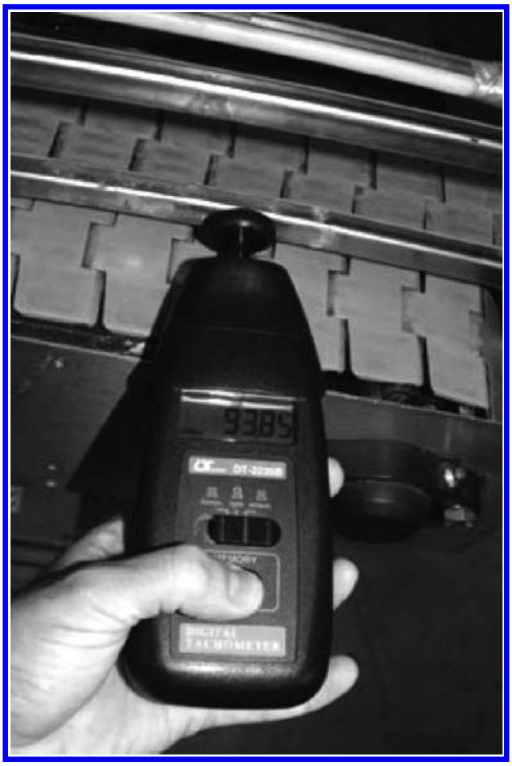


FIGURE 5.64
A digital tachometer is used to measure the velocity of a conveyor belt.

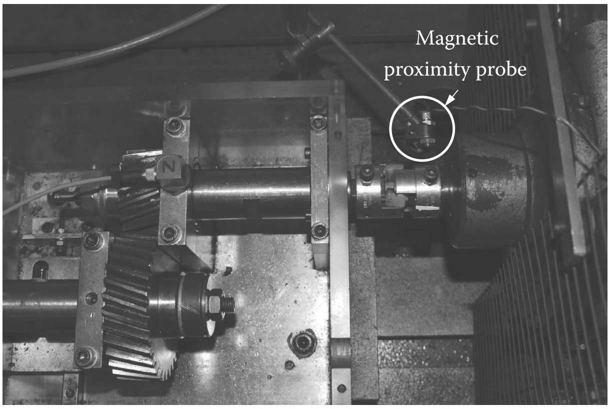


FIGURE 5.65
A magnetic proximity probe is used as a tachometer to measure the angular velocity of a driving shaft in a gear box.

probes can also be used to measure a distance or to just act as a switch. The basic principle assumes that the rotating object must have some type of reference, such as a reflector, magnets, or teeth. In the example case shown in Figure 5.65, the proximity probe used is of the electromagnetic type, operating under the principle of eddy currents, such as many speedometers that are still being used in cars today.

Eddy currents are circular induced currents produced within conductors that can generate their own magnetic fields. Any material that is electrically conductive will conduct an induced current when exposed to a varying magnetic field. The stronger the applied magnetic field and the faster the field changes, the greater the currents that are developed.

Electromagnetic proximity probes are extremely sensitive to small displacements; hence, they can be used to measure either a displacement or a velocity. Most of the proximity probe–based tachometers work the following way: When the shaft in Figure 5.65 is rotating, the protruding head of the steel bolt (which works as a reference) passes in front of the proximity probe every time the shaft completes a full turn. Every time the head of the bolt passes in front of the probe, the magnetic field increases. Based on these changes, a signal processing system counts the number of times the head of the bolt passes in front of the proximity probe per second.

There are, however, many other technologies available, for example, an infrared sensor that uses the reflection of an infrared light in a strip of reflector, to measure rotational speed.

5.5.3 Displacement Transducers

5.5.3.1 LVDT

Linear variable (or voltage) differential transformers (LVDTs) have been used extensively for the accurate measurement of translational displacement and for the control of position within closed-loop systems [31]. When used for vibration measurements, typical units are mm or μm. Similarly, a RVDT (rotational variable differential transformer) is used to measure angular displacements (rotations). LVDTs are robust and durable sensors.

The principal of operation is explained using Figure 5.66. The LVDT is composed of a ferromagnetic core that is placed coaxially to three solenoid coils (one primary at the center and two secondary outer coils). The primary coil is connected to an AC power supply at an

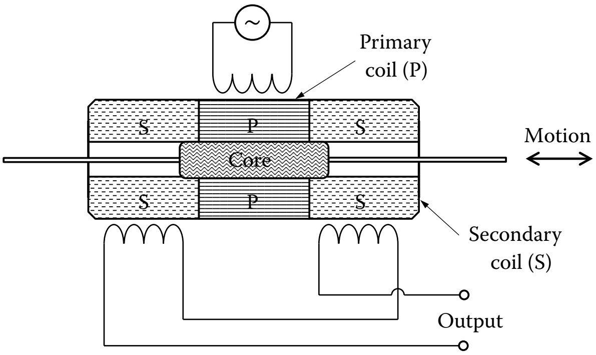


FIGURE 5.66
Schematics of the principle of operation of an LVDT.

excitation frequency of 1 to 10 kHz. This produces an alternating electromagnetic field at the center of the transducer. Depending on the position of the core, this will induce a voltage in the secondary coils. When the core changes position, the induced voltages change. Because the coils are connected as shown in Figure 5.66, this will produce a voltage differential at the output, which will be related to the position of the core. When the core is at the center position, this differential should be zero (in principle), due to symmetry. When the core occupies the extreme left or right positions, this differential reaches a maximum value. However, the differential has a signal, given by the phase between the output and the primary coil current. If the differential is determined from the difference between the left secondary coil and the right secondary coil, in this order, when the core moves to the left, the differential will grow from zero, and its value is positive whereas when the core moves to the right the differential will decrease from zero, and its value is negative. This way, it is possible to know, exactly, the position of the core inside the transducer.

There is a clearance between the core and the coil so that, when in operation, the core does not touch the coil, and thus motion is frictionless. This makes of the LVDT a very reliable and durable sensor with virtually no wear when properly used.

Figure 5.67 shows an example of application of an LVDT to measure the displacement of a SDOF system when under free vibration.

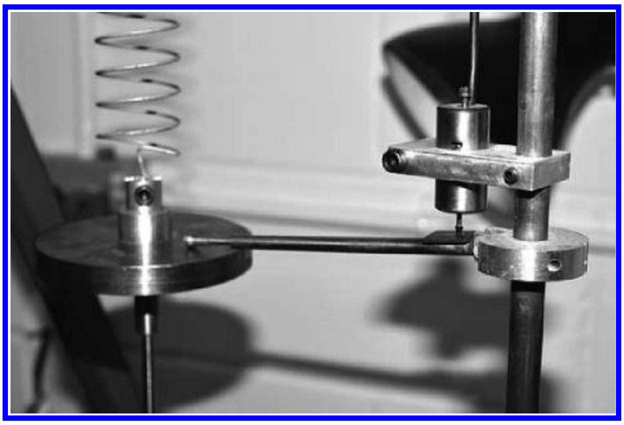


FIGURE 5.67
Example of application of an LVDT to measure the displacement of a SDOF free vibrating system. The LVDT is the cylindrical sensor on the right.

5.5.3.2 *Laser*

Lasers can also be used for contactless measurement of displacement or position and not only velocity. In this case, the principle of operation has nothing to do with the Doppler effect as in LDVs. Instead, they are based on the principle of triangulation of light.

The principle of triangulation is illustrated in Figure 5.68: The laser beam reflected by the structure passes through a receiver's lens and hits a CCD (charge-couple device) sensor. The CCD is composed by a grid of photo detectors, such as photodiodes, that are able to convert light into voltage.

These laser devices can be extremely sensitive: Some are even used for checking and mapping roughness. Other applications include micropositioning, semiconductors, silicon wafers, lenses or circuitry production, vibration measurement, and differential thickness measurement.

Figure 5.69 shows an application in which a FLW laser nanosensor is being used to determine the displacement at the tip of a specimen being tested in a very high cycle

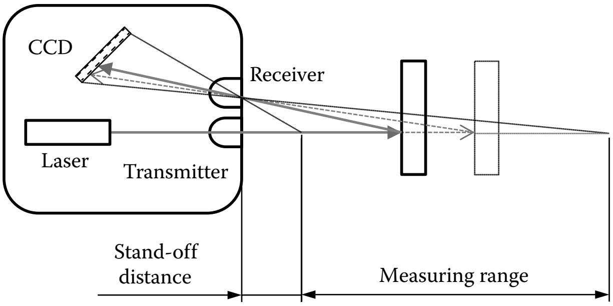


FIGURE 5.68
Principle of triangulation of laser light for the measurement of displacement.

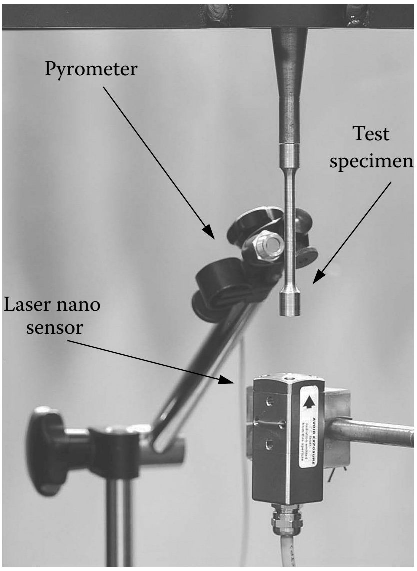


FIGURE 5.69
Using a laser nanosensor to measure displacement at the tip of a specimen in a very high cycle fatigue (VHCF) machine.

fatigue (VHCF) machine [32]. In a VHCF test, the specimen is designed to have the same fundamental axial frequency as the excitation source. In the case shown, the fundamental axial frequency of the specimen is 20 kHz. Because the temperature generated at the center of the specimen will increase dramatically, a pyrometer is used to monitor temperature and trigger a cooling function whenever temperature increases above a certain limit. The abrupt changes in temperature make strain gauges unreliable because temperature affects their resistance. Also, their life is shorter than the specimen being tested and will eventually stop functioning before the end of the test. As an alternative, a laser nanosensor is used to measure the displacement at the tip of the specimen, which can be related to the strain in the middle of the specimen, and thus stress, using analytical equations [32].

The laser nanosensor shown in Figure 5.69 has a working frequency up to 200 kHz and has an output response that is approximately linear up to 5 V. The standoff distance is 1.02 in (26 mm), meaning that the distance between the laser and the measurement target must be fixed and precise within a tight tolerance for linear output, contrary to a laser vibrometer with which distance to the target is immaterial (as long as focus is obtainable). The maximum measurement range for linear output results is 0.13 mm, but the resolution is extremely fine (22.1 nm). The major inconveniences with the use of such a sensitive technology is that the measurement range is somewhat limited, the measurement surfaces must be very well polished (mirror like), and the surface must be perpendicular to the laser beam path.

5.5.3.3 Proximity Probes

Proximity probes can be used to measure not only velocity, but also a distance or the presence of an object that crosses the line of action of that sensor. This is because, above all, a tachometer of the proximity probe type is a counter. If the sensor is of the electromagnetic type, working under the eddy currents principle, it will be highly sensitive to a change in the magnetic field, which can be related to the distance between the probe and the object (as long as the object is electrically conductive).

Figure 5.70 shows an infrared proximity probe that is used to detect the presence of bottles in a conveyor. The proximity probe has both a transmitter and a receiver. It sends a beam of infrared light that, once reflected by an object, is detected by the sensor. As long

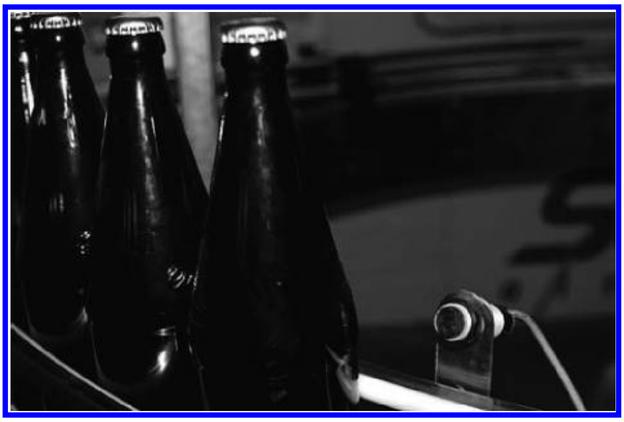


FIGURE 5.70
An infrared proximity probe is used to detect the passage of bottles in an industrial bottle washing machine.

as the target is within a certain distance, called *nominal distance*, the sensor is capable of detecting the passage of any infrared reflective object.

In the case of a bottle of glass, an electromagnetic sensor would not be suitable. The sensing technology in which the proximity probe bases its operation depends on some properties of the target as well.

5.5.3.4 MEMS Sensors

Micro electromechanical system (MEMS) sensors are smaller versions of some of the previous sensors. They have been gaining popularity in recent years. They exist in many consumer products today, for example, smartphones, game consoles, digital cameras, GPS devices, tablets, etc.

MEMS *accelerometers* operate under the same principles as capacitive and piezoresistive accelerometers (see Section 5.5.1). As with the capacitive accelerometers, they can measure accelerations down to 0 Hz (statics). This means that they can sense the acceleration of gravity, and thus orientation with respect to Earth, or a centripetal force in uniform circular motion, which conventional piezoelectric accelerometers used for sensing vibrations cannot.

MEMS *gyros* are capable of measuring angular velocities around one or more axes. They are many times confused with gyroscopes, which is a different type of sensor. MEMS gyros are based on the Coriolis effect rather than on a gyroscopic reaction of a rotating body. The Coriolis and gyroscopic effects are different physical phenomena. The Coriolis acceleration appears in bodies that are moving at the radial direction of a rotating body (as it happens on Earth when an aircraft travels from south to north and vice versa) whereas the gyroscopic reaction occurs as a consequence of the conservation of the angular momentum when the axis of a rotating body changes direction (as happens on a bicycle, thus giving the rider balance).

Figure 5.71 depicts the principle behind a single-axis MEMS gyro of the tuning fork type. Two masses are kept vibrating in opposite directions with velocity v. When a rotation

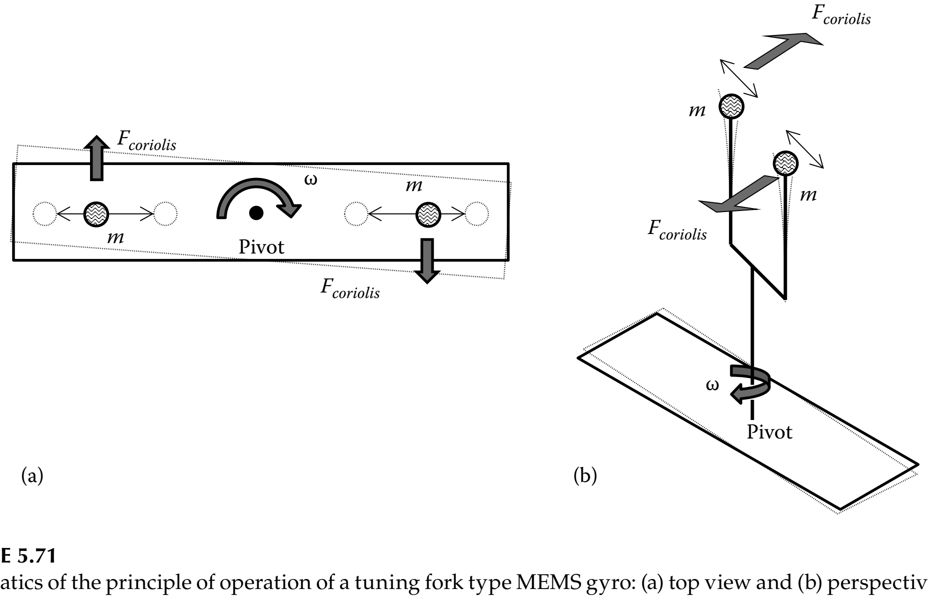


FIGURE 5.71
Schematics of the principle of operation of a tuning fork type MEMS gyro: (a) top view and (b) perspective view.

about the pivot point, with angular velocity ω, is produced, the fact that the masses are moving at radial directions with respect to the rotational axis will create a Coriolis acceleration on each one of the two masses. From Newton's second law, it is possible to derive a force proportional to the mass m as

$$\boldsymbol{F} = 2m\boldsymbol{\omega} \times \boldsymbol{v} \tag{5.96}$$

where × denotes the vector cross-product. The Coriolis force generated is enough to displace the masses from their equilibrium position. Because these masses are moving electrodes that are placed between fixed electrodes, a change in the electrical capacitance proportional to displacement is induced.

5.5.4 Strain Gauges

The measurement of strain can be particularly useful in many applications. For example, it can be used to determine critically loaded (stressed) locations in a structure as illustrated in Figure 5.72.

Another example of the importance of the measurement of strain is in uniaxial tensile testing, in which a specimen is stretched (or compressed) along one direction. The measurement of strain provides us with information necessary to determine some of the materials' properties. *Engineering constants* (sometimes known as *technical constants*) are generalized Young's moduli, Poisson's ratios, shear moduli, and some other behavioral constants. These constants are measured in simple tests, such as uniaxial tensile or pure shear tests [33]. Most simple material characterization tests are performed with a known load or stress. The resulting displacement or strain is then measured.

Strain can be measured directly with a *strain gauge* (Figure 5.73a) or with an *extensometer* (Figure 5.73b). When measuring with an extensometer, strain is determined as the ratio between the displacement $\Delta l = l - l_0$ and initial length l_0 between two marks, that is,

$$\varepsilon = \frac{l - l_0}{l_0} = \frac{\Delta l}{l_0} \tag{5.97}$$

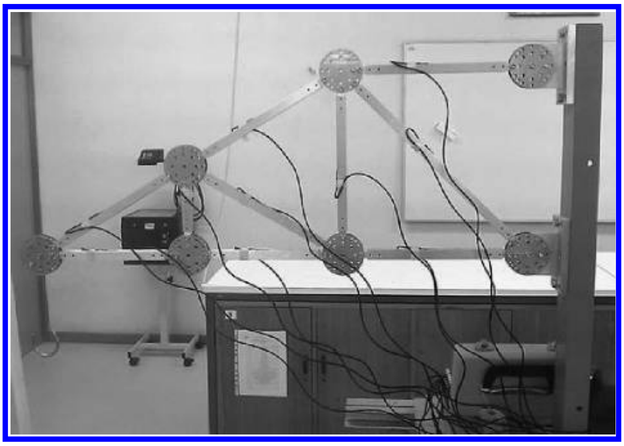


FIGURE 5.72
Measuring strains in a scale model of a trussed structure to determine the most critically loaded members. (Photo taken at the Polytechnic Institute of Setubal in Portugal.)

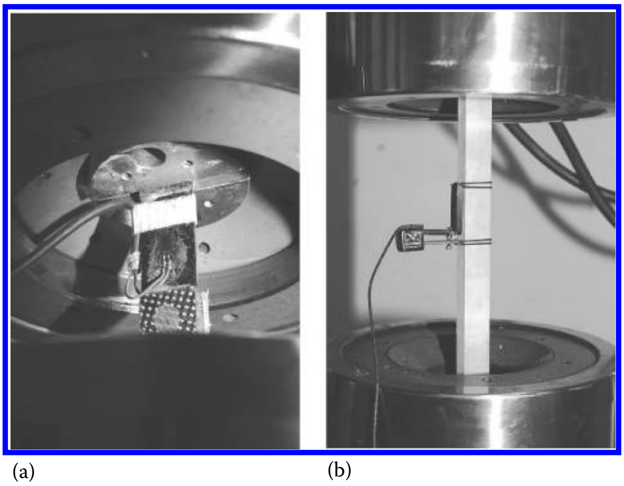

(a) (b)

FIGURE 5.73
Measuring strain in uniaxial test hydraulic test machines (a) with strain gauge and (b) with clip-on extensometer.

The engineering constants are generally the slope of a stress σ versus strain ε curve, that is, the Young's modulus:

$$\sigma = \frac{F_1}{A} = E\varepsilon \tag{5.98}$$

or the slope of a strain–strain curve, that is, the Poisson's ratio:

$$\nu = -\frac{\varepsilon_2}{\varepsilon_1} \tag{5.99}$$

for $\sigma_1 \neq 0$, all other stresses are zero, and $\varepsilon_2 \perp \varepsilon_1$. In these equations, F_1 is the force (measured by a load cell, described in Section 5.5.5, along the specimen's longitudinal direction), and A is the specimen's cross-sectional area at the location at which the strain is placed. Figure 5.74 shows two mutually orthogonal strain gauges, used to determine the Poisson's ratio through Equation 5.99. However, multiaxial strain gauges can be found for this purpose as well: These are called *rosettes*, consisting of two or more strain gauges

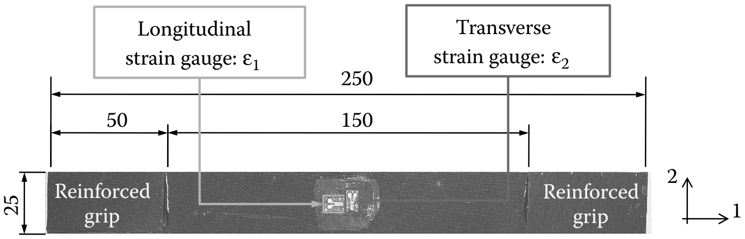

Longitudinal strain gauge: ε_1
Transverse strain gauge: ε_2
250
50
150
25
Reinforced grip
Reinforced grip
2
1

FIGURE 5.74
A carbon fiber composite specimen, manufactured according to ISO 527-4:1997 (type 2), is instrumented with two orthogonal strain gauges for the measurement of the Young's modulus and Poisson's ratio.

arranged at different directions (typically 0°, 45°, and 90°). A few examples of strain gauges are shown in Figure 5.75.

The strain gauge is generally much more accurate than the clip-on extensometer. However, the process of gluing a strain gauge and wiring the leads is very time-consuming as it is a work of minutia, requiring expertise. Also, strain gauges respond to temperature changes: In fact, temperature-related effects are among the major sources of error in strain measurements. This is because most strain gauges are made from metallic alloys, the electrical resistivity of which changes with temperature. This important property is in fact taken as an advantage by some temperature sensors (see Section 5.5.6).

In terms of principle of operation, a strain gauge is an electrical circuit the resistance of which changes with strain. The most popular circuit used in strain sensing is called the *Wheatstone* bridge, a network of resistances named after Sir Charles Wheatstone.

To better understand how strain gauges work, it is better if this is explained using the quarter bridge as an example (Figure 5.76), with which the resistances R_1 and R_2 are equal, and the resistance R_{SG} is variable (representing the strain gauge). The resistance R_3 represents a rheostat, which is usually set at a value equal to the strain gauge resistance when no force is applied. Typically, commercial strain gauges have one from two resistances (when at rest): 350 Ω or 120 Ω.

When a load is applied, the resistance R_{SG} on the strain gauge changes: It increases when it is stretched, and it decreases when it is compressed. With reference to the

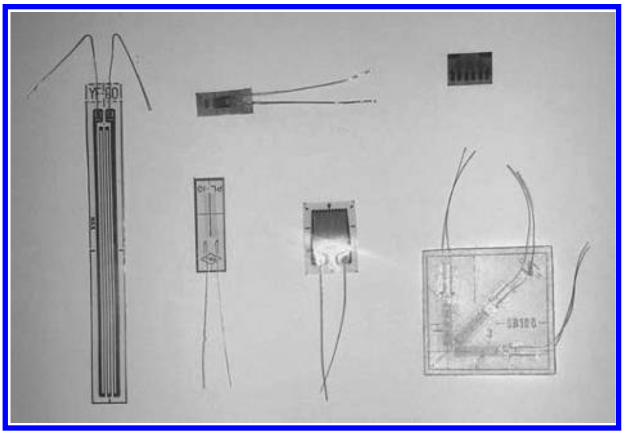


FIGURE 5.75
Different strain gauges. The two on the right are called *rosettes*.

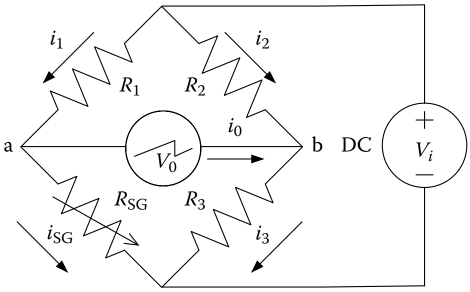


FIGURE 5.76
Schematics of the Wheatstone bridge (quarter bridge).

schematics illustrated in Figure 5.76, it should not be hard to show that the output voltage in a Wheatstone bridge is

$$V_o = \left(\frac{R_1}{R_{SG} + R_1} - \frac{R_2}{R_3 + R_2} \right) V_i \tag{5.100}$$

The bridge is said to be at *balance* when unloaded, that is, when the current is $i_o = 0$ and the voltage is $V_o = 0$ when measured between points *a* and *b*. Thus, in a balanced bridge,

$$\frac{R_1}{R_{SG}} = \frac{R_2}{R_3} \tag{5.101}$$

It should be easy to extrapolate these two equations for the half bridge and full bridge configurations. In a half bridge, two legs have variable resistance, and in a full bridge, all the legs have variable resistance (Figure 5.77). Opposing legs should be stressed in different directions (when one is being stressed, the opposing one should be being compressed).

One example on the use of a half bridge is when measuring a beam subjected to deflection, for example, as in the piezoresistive accelerometer shown in Figure 5.61 earlier. In a half bridge configuration, strain gauges are attached at different sides of the beam so that when one side is under traction, the opposite side is under compression. This is illustrated in Figure 5.78. With respect to the full bridge, this is exemplified later on when discussing load cells' design (see Section 5.5.5).

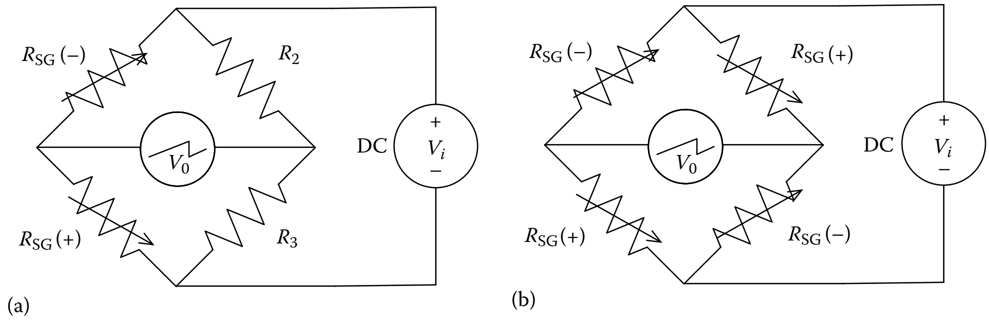


FIGURE 5.77
Schematics of the Wheatstone bridge: half bridge (a) and full bridge (b). The signals point out that when one member is stretched (+) the other should be compressed (−) for the bridge to be used in strain sensing.

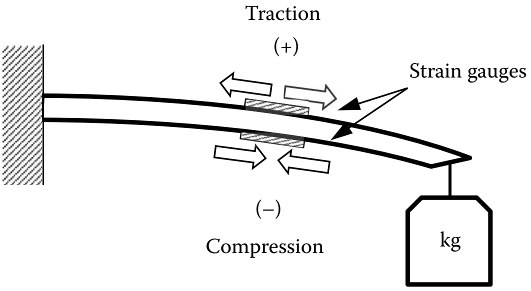


FIGURE 5.78
Example of application of the half bridge.

Even if both the half bridge and full bridge configurations offer better sensitivity, the quarter bridge is the most used configuration because it is the simplest one in terms of setup. However, the quarter bridge configuration, as well as the half bridge, can be highly nonlinear, offering a narrower strain measurement range. On the contrary, the full bridge configuration output voltage V_0 is directly proportional to the applied force as long as the force produces an equal change in resistance in all four legs.

5.5.5 Load Cells

The same type of technology used in transducers that measure quantities, such as displacement, velocity, acceleration, strain, etc., can also be seen in transducers that measure force, that is, load cells. Available technologies include piezoelectric, capacitance, strain gauge, electromagnetic, tuning fork, etc.

In this section, the two most popular technologies will be discussed: piezoelectric and strain gauge–based.

5.5.5.1 Piezoelectric Force Transducers

Piezoelectric force transducers operate under the same principle as piezoelectric accelerometers although force transducers do not have a seismic mass. When the piezoelectric crystal is deformed, a charge output, proportional to the rate of change of the force acting on the crystal, is produced. As piezoelectric accelerometers, force transducers can either be of the type charge or IEPE, depending if they bring built-in pre-amplifiers or not. Most piezoelectric force transducers do not measure static (DC) forces.

The way a force transducer is constructed reminds one of the way a sandwich is made: A piezoelectric crystal is placed between a base case and a top case. Usually, one side in this "sandwich" is much lighter than the other. In a conventional setup, the force transducer is placed with the lighter side (called *base side*) toward the structure and the heavy side away from the structure (Figure 5.79a) [6]. This is done in order to avoid as much modification to the structure as possible. The mass of the side that is attached to the structure is called *active* or *live mass*. This is the mass that is "seen" by the structure in the sensing direction. However, at perpendicular directions, the structure will "see" the total mass of the force transducer. These masses have been reported to be 3 g on the base side for a conventional force transducer weighing a total of 21 g although this varies from model to model [6].

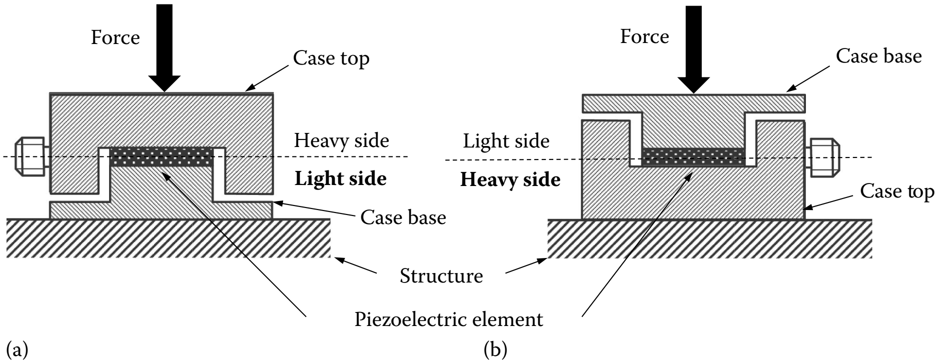


FIGURE 5.79
Schematic cross-sectional view of a piezoelectric force transducer: (a) conventional mounting and (b) upside-down mounting.

Nevertheless, the force transducer can be attached with the heavier side on the side of the structure (Figure 5.79b). In this case, the discrepancy between the active mass (18 g) and the mass "seen" by the structure in perpendicular directions (21 g) will be smaller.

One particular example in which force transducers are used is in the measurement of the FRF of a structure. The FRF contains information about the natural frequencies, modal damping factors, and mode shapes of a structure. For a harmonic excitation, the FRF is the relationship between the response and the force. If an accelerometer is used and as long as the excitation is harmonic, the FRF at a given frequency ω is

$$\alpha(\omega) = \frac{\ddot{x}(t)}{F(t)} \tag{5.102}$$

where $\ddot{x}(t)$ is the acceleration response to the input force $F(t)$. Because this is a representation of the FRF that makes use of the acceleration, $\alpha(\omega)$ is called *accelerance*. If the FRF was determined from the velocity or displacement responses, it would be called *mobility* or *receptance*, respectively. Again, as explained when introducing the FRF with Equation 5.70, it is important to emphasize that Equation 5.102 cannot be used unless the signals are harmonic.

Most of the time, it is not practical to measure the FRF because the excitation force is unknown. However, when in a laboratory environment, it is possible to generate an excitation function, the force of which is transmitted to the structure using a shaker (Figure 5.80). Typically, these excitation functions are of the type random, pseudorandom, sweep sine, multisine, or stepped sine [34]. When using a synchronous excitation of the type sweep sine, multisine or stepped sine, a uniform window can be used, and no leakage is expected to occur.

Push rods (also referred in the literature as *stingers* or *drive rods*) are used to apply the excitation force from the shaker to the structure. The objective is to transmit controlled excitation to the structure in a given direction and, at the same time, to impose as little constraint on the structure as possible in all other directions. The locking ball joint fixture allows for simple alignment of the excitation direction, when the structure is not exactly perpendicular to the exciter's axial direction, also minimizing push rod bending. The whole setup is completely removable and replaceable, thereby avoiding damage to the shaker, structure, or transducers while repositioning is done. One consideration is that it is desirable that the push rod is as stiff as possible in the longitudinal direction so that

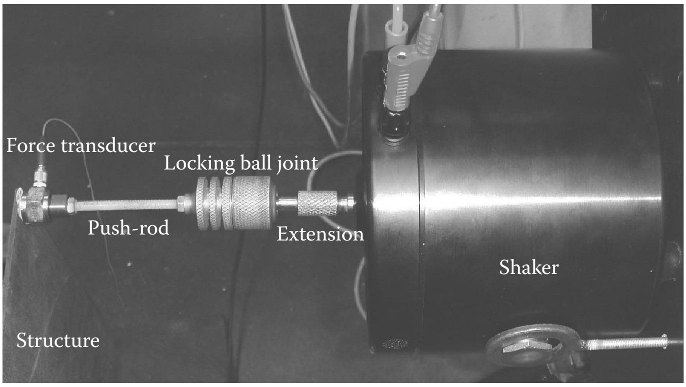


FIGURE 5.80
Push rod connection between a shaker and a force transducer.

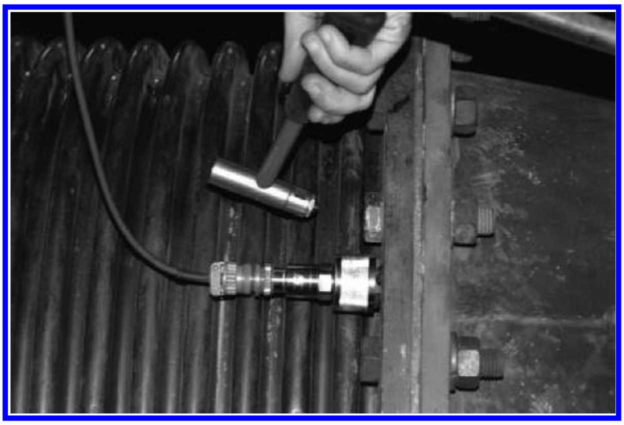


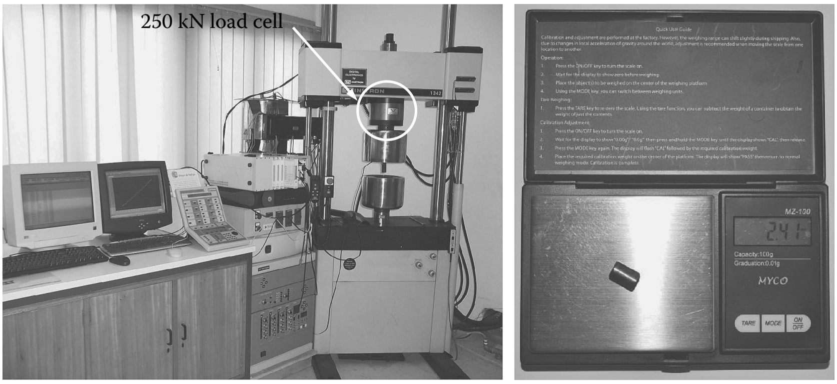
FIGURE 5.81
Using a force transducer instrumented hammer and an accelerometer to measure the natural frequencies at the flange of a 700 mm diameter exhaust pipe in a power plant.

its first mode is above the frequency range of interest but relatively flexible to lateral and rotational motions between its ends.

When it is not practical (or possible) to use a shaker to excite the structure, a hammer can be used instead. In this case, instead of a FRF, one obtains an IRF, the important results of which were introduced earlier in Section 5.4.4. The hammer may or may not be instrumented with a force transducer. An example in which an instrumented hammer is used to excite a structure and thus measure its natural frequencies is shown in Figure 5.81. The antileakage window recommended, in this case, is of the type exponential.

5.5.5.2 *Strain Gauge–Based Load Cells*

Strain gauge–based load cells are used in, for example, universal test machines or digital weight scales, among others. Contrary to piezoelectric force transducers, they can usually measure static (DC) forces. Examples of load cells are shown in Figure 5.82.



(a) (b)

FIGURE 5.82
(a) An Instron hydraulic test machine with a 250 kN load cell (Photo taken at the Polytechnic Institute of Setubal in Portugal). (b) An inexpensive digital pocket scale with a resolution of 0.01 g is being used to measure the weight of a roller bearing cylinder.

Strain gauge–based load cells are made from a precision elastic element that has been manufactured with tight tolerances. This element is often made from aluminum or stainless steel. In a strain gauge load cell, four strain gauges are attached at precise locations on the element and connected at a full bridge Wheatstone configuration (see Section 5.5.4 for more information on bridge configurations). A typical binocular type of elastic element used in strain gauge load cells is illustrated in Figure 5.83. In terms of loadings and boundary conditions, it can be modeled as a cantilever beam with an applied load at its tip. A photo of a load cell with a slight different design is shown as an example in Figure 5.84.

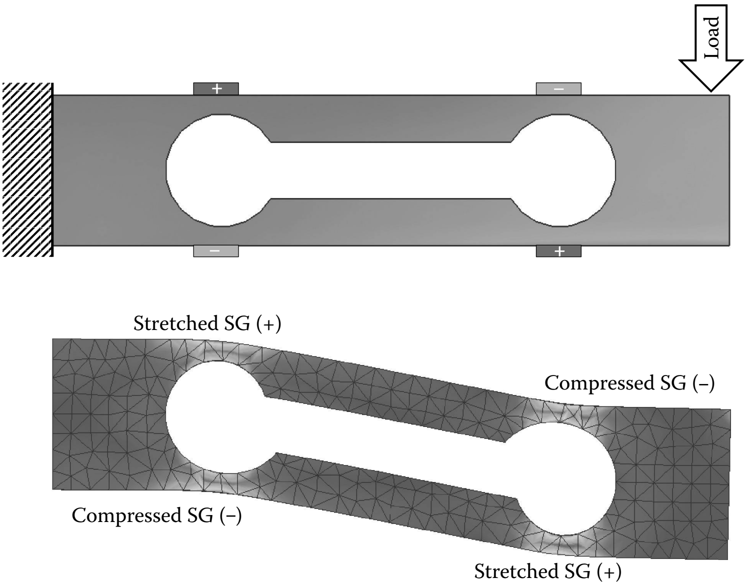


FIGURE 5.83
The finite element model here shows that an elastic element used in strain gauge–based load cells deforms in such a way it makes it possible to have opposing legs in a full bridge Wheatstone configuration to be stressed at different directions. In this picture, SG means strain gauge.

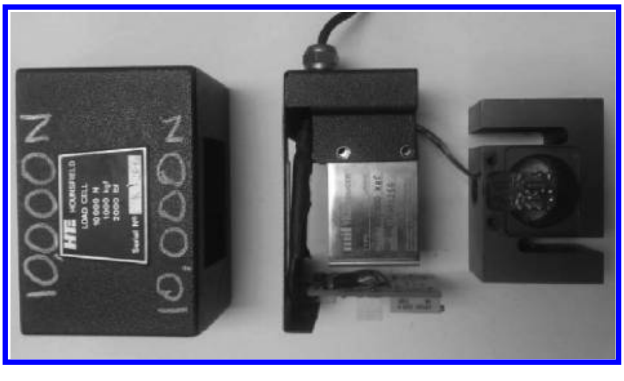


FIGURE 5.84
A stripped-down Hounsfield load cell.

5.5.5.3 Calibration of a Pair Force Transducer versus Accelerometer

Let us recall the equation for the frequency response function expressed in terms of accelerance (Equation 5.102). Using the same notation, Newton's second law can be written as

$$f(t) = m\ddot{x}(t) \tag{5.103}$$

and the FRF given by Equation 5.102 becomes

$$H(\omega) = \frac{\ddot{x}(t)}{f(t)} = \frac{1}{m} \tag{5.104}$$

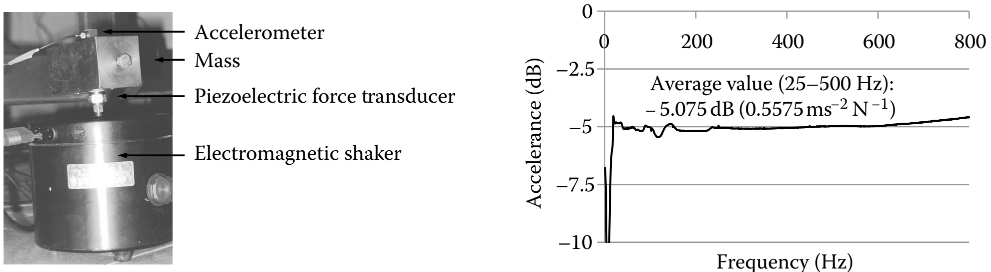
It is very important to note that this Equation 5.104 is only valid within a limited low-frequency range in which the system behaves rigidly. Once the bodies start deforming elastically, it is no longer possible to relate the true value of the mass with the FRF as is suggested in Equation 5.104. This is also why the inverse of Equation 5.104 is often referred to as *apparent mass*.

Taking this into consideration, there is a method that combines the use of an accelerometer with a force transducer (or any other vibration sensor as long as units are converted to acceleration following the procedure explained in Section 5.2.2.3), which allows determining a calibration factor for the pair composed by the force transducer and accelerometer, regardless of their individual calibration values [6]:

$$H_{\text{calibrated}}(\omega) = \Theta\frac{\ddot{x}(t)}{f(t)} = \frac{1}{m_{\text{true}}} \tag{5.105}$$

where Θ is the calibration factor to be determined and m_{true} is the total mass of the system composed by a solid block and sensors. The value of m_{true} can be obtained in a weight scale.

In the example shown in Figure 5.85, the total mass of the system composed by the accelerometer, force transducer, block, glue, and two threaded fixing discs was obtained in a digital weight scale: $m_{\text{true}} = 1.8735$ kg, considering that the force transducer's active



(a) (b)

FIGURE 5.85
Test setup to calibrate the pair force transducer versus accelerometer (a) and plot of the accelerance versus frequency so obtained (b).

mass is 19.73 g [26]. The plot of accelerance versus frequency shows an approximately flat frequency band ranging from 25 to 500 Hz with an average value of 0.5575 $ms^{-2}N^{-1}$. Application of Equation 5.104 to this result yields $m_{\text{uncalibrated}} = 1.7936$ kg. Thus, the calibration factor is determined to be

$$\kappa = \frac{m_{\text{uncalibrated}}}{m_{\text{true}}} = \frac{1.7936}{1.8735} = 0.9574$$

A similar process can be followed to determine the active mass of the force transducer [26].

5.5.6 Temperature Sensors

Temperature sensors are among those that we are most familiar with. These are used in a wide range of different applications: HVAC control, condition and operation monitoring of electronic circuits and machines, food processing, weather forecasting, etc.

In addition to the conventional thermometer that uses the expansion–contraction of mercury in a glass, there are many other different types of temperature sensors: thermocouples, thermistors, resistance thermometers, bimetallic thermometers, and infrared thermometers, among others.

An example of a device that measures both temperature and humidity (a thermo-hygrometer) is shown in Figure 5.86.

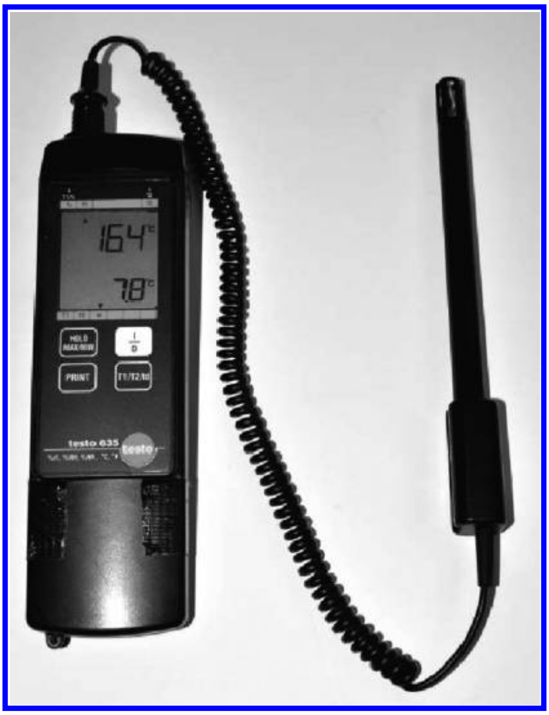


FIGURE 5.86
Photo of a thermo-hygrometer. The value on top (16.4°C) is the room temperature, and the value below (7.8°C) is the dew point.

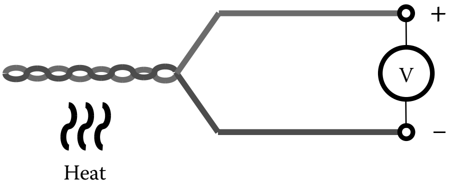


FIGURE 5.87
Schematic of the thermocouple principle.

5.5.6.1 *Thermocouple*

The thermocouple is composed of a pair of dissimilar metals that perform a junction. When dissimilar metals are connected together, an electrical voltage, proportional to temperature, is produced (Figure 5.87). This thermoelectric effect is called the *Seebeck effect* and is named after Thomas Johann Seebeck.

There are different types of thermocouples, made from different alloys. Different alloys determine temperature range, sensitivity, and resolution. The most common one is the K-type, a nickel-based thermocouple. In the K thermocouple, one wire is made from Chromel (nickel–chromium), and the other wire is made from Alumel (nickel–manganese–aluminum–silicone). It has a sensitivity of 41 μV/°C (approximately) and a working temperature range that generally sits between −200°C and 1250°C.

Thermocouples are inexpensive sensors that react rapidly to temperature changes. They do not need any source of energy to work. However, the insulation of the wires degrades with time, and accuracy is affected. They cannot touch other sources of electricity because they are electrical conductors.

5.5.6.2 *Thermistors and Resistance Thermometers*

Thermistors, as well as resistance thermometers or resistance temperature detectors (RTDs), are resistors with which resistance changes with temperature. They can have a positive temperature coefficient or a negative temperature coefficient, depending on whether the relationship between resistance and temperature is direct or inverse. The name "thermistor" comes from the junction between the words "thermal" and "resistor." The major difference between thermistors and RTDs is the material rather than the principle.

A thermistor is made from a semiconductor, such as a polymer, a ceramic, or a metallic alloy (e.g., stainless steel). RTDs are made from pure metals (e.g., platinum), which can come in the form of wire or thin film. RTDs are good for a wider range of temperatures when compared to thermistors from as low as −200°C to as high as 850°C. A thermistor is typically used in more moderate temperature ranges that sit between −90°C and 130°C. However, thermistors can often offer better accuracy over their limited working temperature range.

Like a thermocouple, a thermistor is a readily available and inexpensive temperature sensor. However, they do not have such a quick response to temperature changes although they can deliver more accurate readings.

In a RTD, the change in resistance can be approximated by a linear relationship if the change in temperature is not too large:

$$R = R_0(1 + \alpha T) \tag{5.106}$$

where R_0 is the initial (or reference) resistance, T is the temperature, and α is the temperature coefficient of resistance of the material.

In a thermistor, however, the temperature–resistance relationship is far more nonlinear, in part because the temperature coefficient of resistance of the materials used is usually negative. The temperature–resistance relationship is given by

$$R = R_0 e^{\beta\left(\frac{1}{T}-\frac{1}{T_0}\right)} \tag{5.107}$$

where β, the characteristic temperature (about 4000°K), is temperature-dependent itself, thereby adding to the nonlinearity of the device. Hence, proper calibration is essential, especially when measuring in wider ranges of temperatures (>50°C).

5.5.6.3 Bimetallic Thermometers

When two thin strips of different metals are bound together and their temperature changes, they will expand according to their coefficients of dilatation. However, because they are coupled together, the shear stresses generated will make the two strips bend together in a curved way (Figure 5.88) so that one strip stretches more than the other.

This is the principle behind thermostats that control a room's temperature or dial thermometers used in cooking. In these devices, a bimetallic strip is wrapped into a coil. One end is fixed, and the other end carries a moving electrical contact.

5.5.6.4 Infrared Sensors

Infrared sensors are spectroscopy contactless sensors that can be used to monitor and map surface temperatures. By being contactless, they do not interfere with the structure, which means they do not change their properties (e.g., stiffness or mass when used on small and lightweight components). They can also be used to measure temperature on moving surfaces, on hazardous locations (e.g., high voltage), or on locations of difficult access (e.g., that are too high to reach). Their response to temperature changes is very quick, but their sensitivity depends on the emissivity of the materials (the ratio between the energy radiated by the material and the energy radiated by a true black body at the same temperature). They can measure from as low as –70°C to as high as 1000°C although this depends on the device, which usually operates in much narrower bands of temperatures.

Infrared sensors measure the infrared light that is irradiated from the outer surface of an object. Infrared light has longer wavelength (smaller frequency) than visible light, and it can range from as low as 0.7 μm (≅ 430 THz) to as high as 1 mm (≅ 300 GHz).

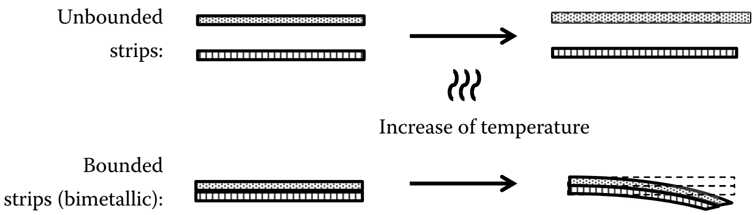
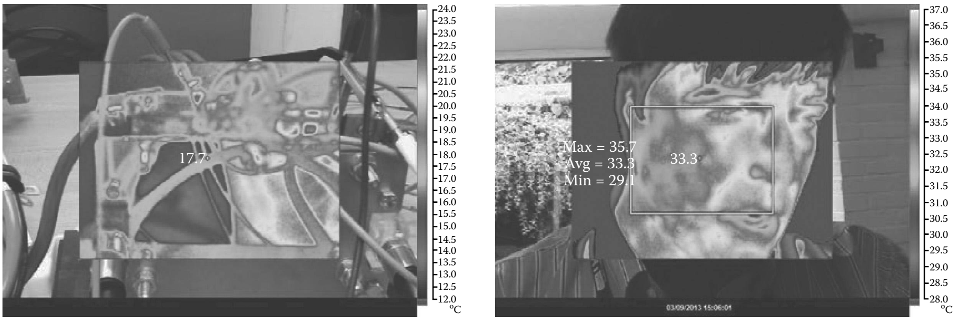


FIGURE 5.88
Bimetallic behavior with temperature increase.



(a) (b)

FIGURE 5.89
Thermal fluid pattern inside a fuel cell (a) (From Scott, P., *Experimental Investigation into a Novel Design Concept for a Modular PEMFC Stack*, PhD Thesis, School of Engineering and Technology, University of Hertfordshire, 2013.) and a picture of this chapter's author (b) taken with a Fluke thermography camera Ti25.

The ISO 20473:2007 standard defines the following infrared spectral bands:

1. Near infrared (NIR): 0.78 μm to 3 μm;
2. Mid-infrared (MIR): 3 μm to 50 μm;
3. Far infrared (FIR): 50 μm to 1 mm.

The NIR region is also called "reflected infrared" because it requires some source of light (in the same frequency range) to be reflected. Infrared night vision goggles operate in the NIR region. Another example of devices that operate in the NIR region are passive infrared (PIR) sensors, such as motion detectors used to control an automatic light.

The MIR region is called "thermal infrared." In this region, a source of illumination is not required. It is within this range that thermography cameras usually operate. Thermography cameras can be used to map temperatures, for example on an engine, on a computer's CPU, on a fuel cell [35], or in the human body (Figure 5.89).

Thermometers that operate at the lower end of the MIR region or in the NIR region are usually called pyrometers and were originally conceived to detect the temperature of very hot objects (usually incandescent and visible to human eye).

The FIR region is used for other applications rather than to measure temperature. Terahertz electromagnetic devices, for example, FIR lasers or radars, can be used in many applications, for example, in the evaluation of material properties, in NDT (nondestructive testing), in the detection of gas leaks, explosives, chemical or nuclear materials, or in medical imaging.

5.5.7 Flow Sensors

It can be shown that the flow Q across a constriction of area A obeys the relationship [18]

$$Q = c_\mathrm{d} A \sqrt{\frac{2\Delta p}{\rho}} \tag{5.108}$$

in which c_d is the discharge coefficient for the constriction, Δp is the pressure drop across the constriction, and ρ is the density of the fluid. The derived SI units for flow are $\mathrm{m^3 s^{-1}}$ as it expresses the volume of moving fluid per unit time.

There are several methods to measure fluid flow. Usually, sensors cannot measure fluid flow directly. Instead, they derive flow from another physical quantity, such as pressure or velocity. Methods used to measure fluid flow can be classified into the following categories [18]:

1. Orifice flow meters: Pressure is measured across a constriction or opening. Examples include nozzles or Venturi meters.
2. Static-pressure meters: The pressure head is measured, which brings the flow to static conditions. Examples include pitot tubes and rotameters. A Venturi meter can also fit in this category.
3. Flow rate meters: Use a turbine from which a change in the angular momentum is measured.
4. Flow velocity meters: Examples include Coriolis meters, LDVs, and ultrasonic flow meters.
5. Indirect flow meters: These are meters that measure flow from an effect it creates. Hot wire anemometers or magnetic induction flow meters are just a few examples.

5.5.7.1 Venturi Tube

The Venturi effect consists of the reduction of fluid pressure that results from flow through a constriction. This effect is exactly the same as is observed in a funnel and is illustrated in Figure 5.90.

Applying Bernoulli's principle, the pressure drop in the Venturi tube is

$$\Delta p = p_2 - p_1 = \frac{\rho}{2}\left(v_2^2 - v_1^2\right) \tag{5.109}$$

which, once replaced in Equation 5.108, allows measuring the flow rate. The pressure can be measured using the techniques outlined in Section 5.5.8.

5.5.7.2 Pitot Tube

Aircraft use pitot tubes to measure air speed. Pitot tubes only measure flow at a given point of the stream because the velocity is not uniform across the flow section. The Pitot

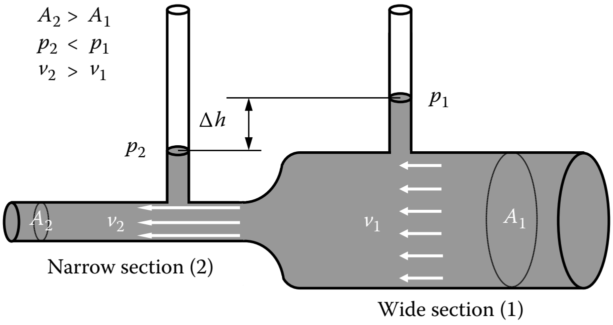


FIGURE 5.90
Venturi tube.

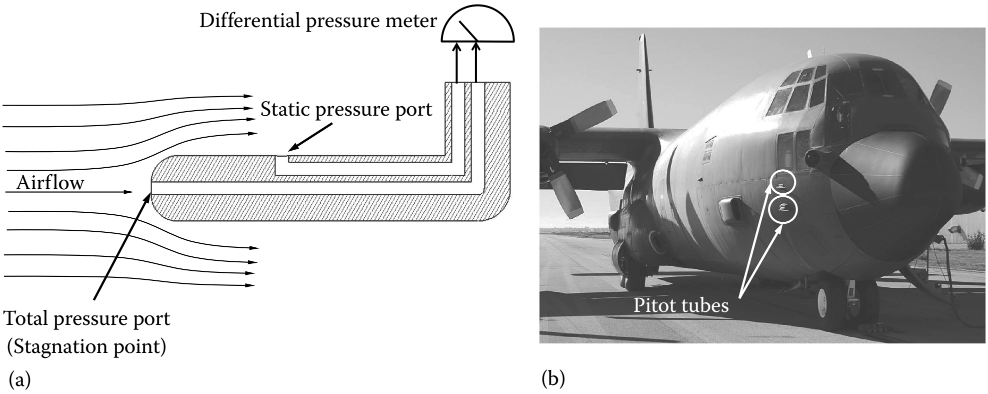


FIGURE 5.91
Pitot tube. (a) Cross-sectional view illustrating the working principle. (b) Pitot tubes on the starboard side of a C-130 Lockheed Martin aircraft.

tube consists of a tube that points directly into the flow at a given location. The moving fluid is brought to rest (stagnates) at the front of the Pitot tube because the tube is already filled in with fluid. This is called the stagnation pressure, total pressure, or pitot pressure. In the pitot-static tube (also known as the Prandtl tube, shown in Figure 5.91), ports are placed radially to the tube to measure the static pressure.

If a fluid is considered to be incompressible (most liquids are incompressible, and gases can sometimes be approximated to behave as incompressible as well) and taking into consideration, again, Bernoulli's principle, the velocity may be determined from

$$V = \sqrt{\frac{2(p_T - p_s)}{\rho}} \tag{5.110}$$

where p_T is the total (stagnation) pressure and p_s is the static pressure.

5.5.7.3 Anemometers and Angular Momentum Flow Meters

An anemometer is a device that is used to measure flow speed rather than flow rate. An example of such a device is shown in Figure 5.92.

In an anemometer, the flow passes through a turbine that is made to spin and is supported in low-friction bearings. The flow speed can be determined once the coefficient of power C_p for the turbine is known. As in a wind turbine, the anemometer will produce a power output P. The flow's velocity v can be determined from

$$C_p = \frac{P}{\frac{1}{2}\rho A v^3} \tag{5.111}$$

where A is the area of the wind turbine.

Another special type of anemometer is the hot wire (or hot film) anemometer. This is usually made from a very thin tungsten or platinum wire. In the hot wire anemometer, a conductor carrying a current i is placed in the fluid flow. Thus, the hot wire is subjected

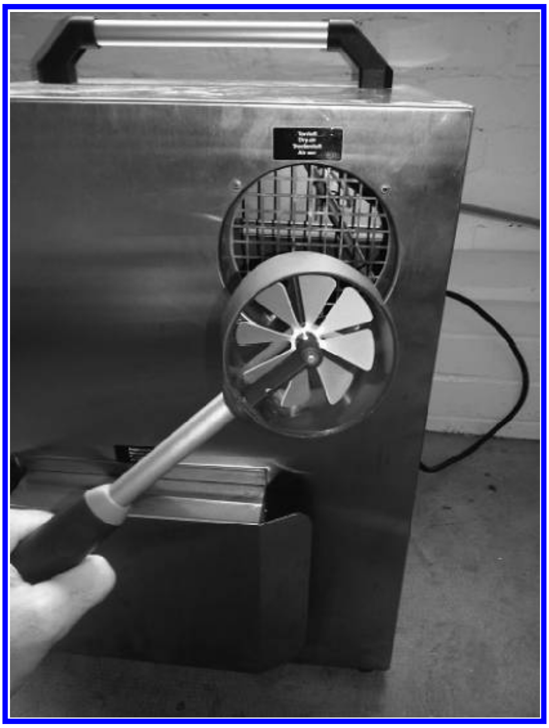


FIGURE 5.92
A portable anemometer is being used to measure airspeed at the outlet of a dehumidifier.

to forced convection. Under steady conditions, the heat loss from the wire into the fluid is exactly balanced by the heat generated by the wire due to its resistance R. Because the coefficient of heat transfer at the boundary of the wire and the moving fluid is known to vary with the square root of the fluid's velocity $\sqrt{v}$, it is possible to relate the velocity of the fluid with the heat balance by

$$i^2R = \left(c_1 + c_2\sqrt{v}\right)(T_w - T_f) \tag{5.112}$$

where c_1 and c_2 are constants that can be determined through a least squares method during calibration (they depend on the geometrical properties of the gauge and physical properties of the medium) [36], T_w is the temperature of the wire and T_f is the temperature of the fluid.

In the angular momentum flow rate meter shown in Figure 5.93, a motor is used to govern a tube bundle through which the fluid flows. The motor torque T and angular speed ω are measured. As the fluid mass passes through the tube bundle, it imparts an angular momentum at a rate governed by the mass flow rate Q_m of the fluid. The motor torque provides the torque needed to balance this rate of change of angular momentum. The governing equation is

$$T = \omega\bar{r}^2 Q_m \tag{5.113}$$

where $\bar{r}$ is the radius of gyration of the rotating fluid mass.

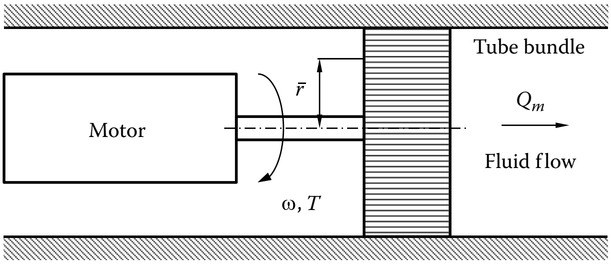


FIGURE 5.93
Angular momentum flow meter.

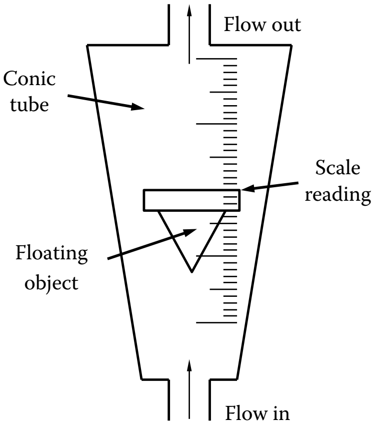


FIGURE 5.94
Rotameter.

5.5.7.4 Rotameter

One well-known sensor is the rotameter shown in Figure 5.94. Basically, it consists of a cylindrical object floating inside a vertical tube with a varying cross-section. Typically, both tube and floating object are conic with the taper growing in the same direction. The weight of the floating object is balanced by the pressure differential on the object. When the flow speed increases, the object rises within the conic tube, thereby allowing more clearance between the object and the tube. The tube is often made from glass and includes a scale, so that the reading can be made directly from the position of the floating object inside the tube.

5.5.7.5 Other Flow Measurement Sensors

The LDV principle used to measure structural velocity as explained earlier in Section 5.5.2 can be applied to fluid flow measurements as well. Once more, one particular advantage of using laser-based technology is that it does not physically interfere with the system, that is, it does not have an effect on the flow.

Another method of sensing velocity in a flow consists of an ultrasonic burst sent in the direction of the flow. The time of flight, that is, the time the burst takes to travel through the medium, is measured. The ultrasonic wave propagations' speed is related to the fluid's velocity.

The Coriolis principle can also used to measure flow. In a method, the flow is made to flow through a U-shaped/parabolic tube. Because the parabola does not have constant radius (as a circle does), an acceleration of Coriolis is generated, which can be sensed by displacement sensors from the lateral motion of the tube.

There are many other direct and indirect methods to measure fluid flow. One important thing to note is that, quite often, flow and pressure measurements can be related to one another, so these sensors can sometimes share similar principles of operation.

5.5.8 Pressure Transducers

As the name suggests, pressure sensors are used to measure pressure, usually exerted by fluids (liquids or gases). They are widely used in aerodynamics, hydraulics, and pneumatics applications. The SI unit is Pa (Pascal), which is equal to Nm^{-2} (force per unit area) even though pressure sensor units often come expressed in bar, atm, mmHg (or torr), psi, etc. Some usual pressure units and their conversion to SI units are shown in Table 5.7.

As with thermometers, pressure sensors come in a wide variety of forms and are present in many everyday applications. Depending on the application, pressure sensors can be used not only to measure pressure as an end in itself, but also flow, airspeed, water level, depth, altitude, or leaks. Another example of application is a binary pressure switch, such as a computer mouse button, a doorbell, or speed detectors in some traffic enforcement cameras.

Almost all pressure sensors are differential in the sense they measure pressure against a reference value whether it is vacuum pressure, atmospheric pressure, or another reference. An example of a differential pressure gauge is shown in Figure 5.95.

The technology used to measure pressure can be very similar to that used in force transducers and load cells. Usually, a membrane, a spring, bellows, a bourdon tube, or a piston are used to measure deflection due to an applied force over a known area. The point is that—most of the time—the deformation of a sensing element can be converted into a reading of the exerted force and, once the area is known, into pressure.

Pressure sensors can be piezoelectric, piezoresistive, capacitive, strain sensitive, potentiometric, electromagnetic, inductive, thermal, or flow sensitive, among others. Most of

TABLE 5.7

Pressure Units Converted to SI Units ($1\ Pa = 1\ Nm^{-2} = 1\ kgm^{-1}s^{-2}$)

Name	Symbol	Conversion to SI Units
Technical atmosphere	at	$1\ at = 0.98067 \times 10^5\ Pa$
Standard atmosphere	atm	$1\ atm = 1.01325 \times 10^5\ Pa$
Barometric pressure	bar	$1\ bar = 1 \times 10^5\ Pa\ (\cong 1\ atm)$
Torricelli	Torr	$1\ torr = \frac{1}{760} atm = 133.3\ Pa$
Height of column of mercury	mmHg	$1\ mmHg \cong 1\ torr = 133.3\ Pa$
Height of column of water	mmH_2O	$1\ mm\ H_2O = 9.8067\ Pa$
	inH_2O	$1\ in\ H_2O = 248.84\ Pa$
Pounds per square inch	psi	$1\ psi = \frac{1\ lbf}{1\ in^2} = 6894.8\ Pa$
Hectopascal/millibar	hPa/mbar	$1\ hPa = 1\ mbar = 1 \times 10^2\ Pa$
Meter sea water	msw	$1\ msw = 0.1\ bar = 1 \times 10^4\ Pa$
Foot sea water	fsw	$1\ fsw = 3.0643 \times 10^3\ Pa$

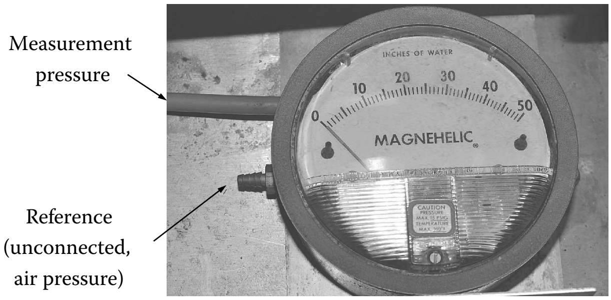


FIGURE 5.95
A differential pressure gauge.

these technologies have been discussed in the previous sections even if they were exemplified under the context of different applications.

For example, it should already be easy to understand that piezoelectric pressure sensors are better suited for dynamic measurement (high-speed changes in pressure) as it happens inside the combustion chamber in an engine. On the contrary, inductive pressure sensors are not as suitable to quick changes in pressure but can be very sensitive and accurate. A flow-sensitive pressure sensor (such as a Pitot tube) is better suited to measure airspeed in an aircraft.

5.5.9 Ultrasonic Sensors

Ultrasonic sensors use pressure waves that are transmitted and reflected at frequencies above the human audible range. Ultrasonic sensors are used in many applications, including medical imaging, nondestructive testing, flow measurements, sonars, burglar alarms, and car parking sensors. Ultrasonic equipment can use frequencies as low as 20 kHz and can go up to as high as the GHz region (1 GHz = 1×10^9 Hz).

Ultrasound can be generated according to several principles, for example, from very high frequency oscillations on a piezoelectric element subjected to an electrical potential or from the use of the magnetostrictive property of ferromagnetic materials when subjected to oscillating magnetic fields.

In essence, ultrasonic sensors project an ultrasonic burst toward a target object, and the time taken for the echoes to be received are clocked. A signal processor, which is calibrated according to the speed of sound in the medium in which the sound propagates, determines the position of the echoes with respect to the probe. In other words, position can be determined from

$$x = \frac{ct}{2} \tag{5.114}$$

where c is the speed of sound, and t is the time of flight (from generator, then to target, and finally to receiver).

As an example, a C-scan used to map voids or delamination defects throughout the thickness of composite laminates is illustrated in Figure 5.96. In this picture, the times of flight t_1, t_2, t_3, and t_4 are relative to the echoes that happen every time there is a transition: t_1 is the echo at the entrance of the laminate, t_2 is the echo at the defect, t_3 is the echo at the exit of the laminate, and t_4 is the echo at the bottom of the tub.

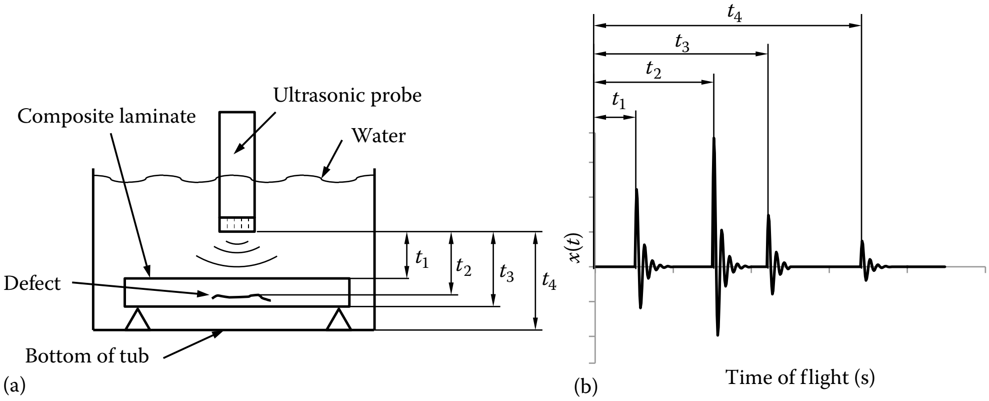


FIGURE 5.96
Schematics of the principle of operation of a C-scan used for mapping damage in composite laminates (a) and echoes seen in the time domain (b).

Alternatively, the velocity of the target (if moving) can be measured using the Doppler effect, a principle that has been described before in Section 5.5.2, although using a different sensor as an example.

5.5.9.1 Encoders

Encoders are a specific class of digital transducers that can monitor motion or position from a coded (digital) reading of a measurement. In general, encoders are found in the form of discs that display a code pattern. These can be divided into two classes: incremental encoders (or relative encoders) and absolute encoders.

Encoders are used in a wide variety of applications because they can be used as feedback devices for speed or position control. Applications include CNC machines, automatic welding in an assembly line, elevators, bar code readers, radars, wind turbines, conveyors, printers, robotics, etc.

5.5.9.2 Incremental Encoders

The electromagnetic pulse tachometer shown in Figure 5.65 is an example of a simple type of an incremental encoder: When the ferromagnetic screw passes in front of the sensor during shaft rotation, it counts a pulse (1) whereas it does not pick up any signal (0) when the screw is at any other position. When the shaft completes a full rotation since the last pulse, it will count another pulse (1). The time elapsed between these two pulses is the period, the inverse of which is the frequency in cycles per second (Hz). This can be seen as a type of an incremental encoder.

There are two possible additional configurations for incremental encoders: (1) offset sensor configuration and (2) offset track configuration. When any of these two configurations is used, it is possible to determine both direction (clockwise or counterclockwise) and speed.

Figure 5.97a shows the first configuration. The disk has a single circular track with identical and equally spaced transparent windows. Two photodiode sensors (pick-offs) are positioned facing the track at a quarter pitch (90° or quadrature, i.e., half the window length) apart. Assuming the disk is rotating in the counterclockwise direction at the position shown in Figure 5.97a, the sensor pick-off 1 is changing state (from 1 to 0) at the edge between the

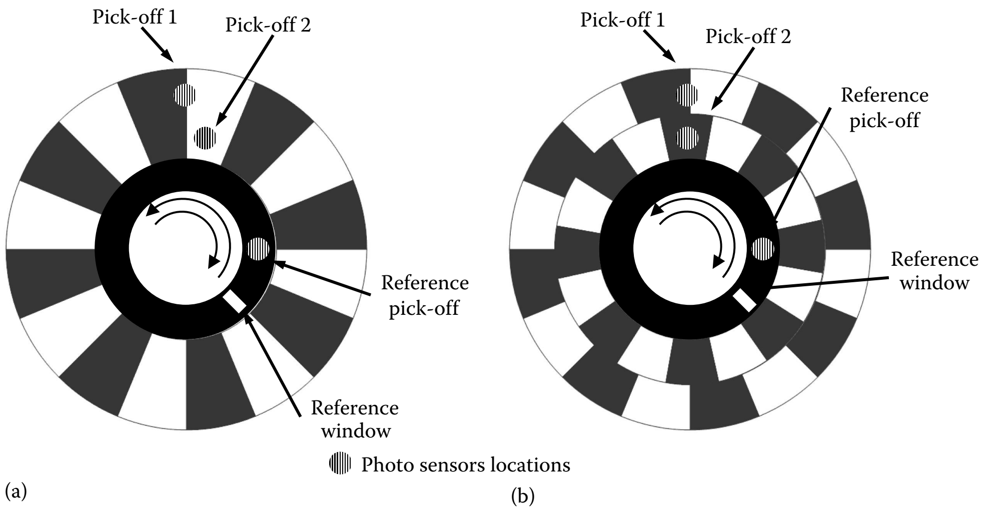


FIGURE 5.97
Incremental encoders: (a) one-track wheel with offset sensors and (b) two-track quadrature wheel.

transparent and opaque windows while sensor pick-off 2 keeps its value unchanged. On the contrary, if the disk is rotating in the clockwise direction, the sensor pick-off 1 will change state from 0 to 1 instead. Encoders must be programmed so that they are able for edge detection as it is based on the sequence of edge detection that the direction can be determined.

The second possible configuration is shown in Figure 5.97b. In this case, the disk has two identical tracks, one offset from the other by a quarter pitch. The two pick-off sensors are aligned along the same radial line unlike the previous configuration. The principle is exactly the same as for the first configuration.

It is not unusual to add an additional track with a single window and associated sensor. This generates a reference pulse (called index pulse) per revolution of the disk, exactly the same way as in the tachometer example shown in Figure 5.65. This index is used as a counter of the complete number of revolutions. When the disk rotates at constant angular speed, the pulse width and pulse-to-pulse period are constant with respect to time. When the disk is accelerating, the pulse width decreases continuously and vice versa.

5.5.9.3 Absolute Encoders

An absolute encoder has many pulse tracks on its transducer disk. The pulses constitute words, written using a number of binary digits (0 or 1) equal to the number of tracks. Typically, the disk has transparent windows for a binary digit of 1 and black opaque windows for a binary digit of 0, exactly the same way as in incremental encoders. A typical encoder uses optical sensors that are capable of differentiating between these windows. Absolute encoders can have different tracks as well as several pulses per track. When the disk of an absolute encoder rotates, several pulse trains—equal in number to the tracks on the disk—are generated simultaneously. Hence, absolute encoders have a system of coded tracks in which no two positions are identical. They also have memory as they do not lose position after power is switched off.

A simplified code pattern on the disk of an absolute encoder, which uses the direct binary code, is shown in Figure 5.98a. The number of tracks n in this case is four, but in practice, n is on the order of 14 and may even be as high as 22 [18]. The disk is divided into 2^n sectors or

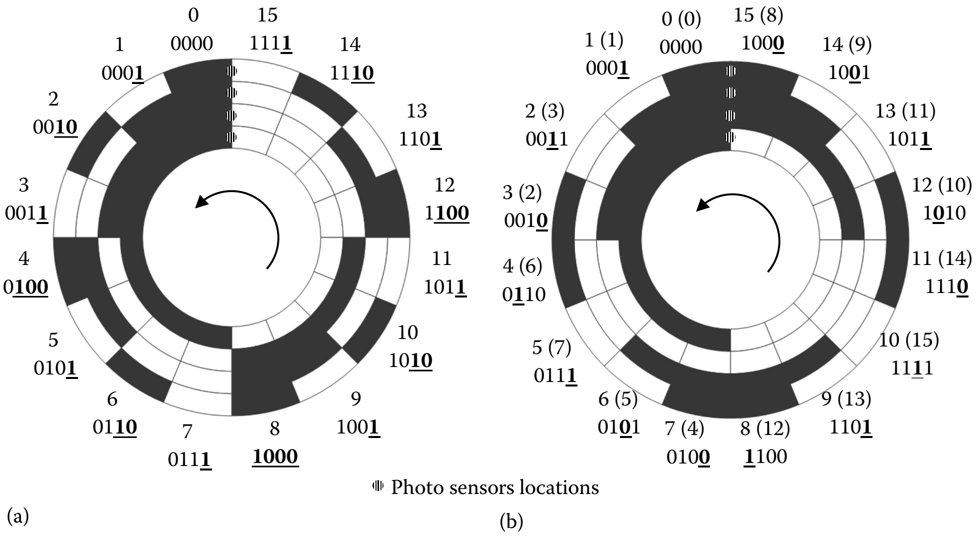


FIGURE 5.98
Four-bit absolute encoders (16 state): (a) binary code and (b) gray code. Bold underlined digits represent a change from the previous state. The numbers between parenthesis () on the gray code represent the position of the word on the binary code.

bits of data. A set of *n* pick-off sensors is arranged along a radial line and facing the tracks on one side of the disk, opposite to a light source that illuminates the other side of the disk. As the disk rotates, at a given instant, the pick-off sensors will generate a combination of signals (coded data word) that uniquely determines the position of the disk at that time.

Straight binary code in absolute encoders can lead to a data interpretation problem. Note that in Figure 5.98a the transition from one sector to an adjacent one may require more than one switching of bits in the binary array of data. This is highlighted with the underlined bold digits that changed from one sector to the other, assuming the disk is rotating in the counterclockwise direction. For example, the transition from sector 7 (0111) to sector 8 (1000) requires four-bit switching, the transition from sector 9 (1001) to sector 10 (1010) requires three-bit switching, and the transition from sector 11 (1011) to sector 12 (1100) requires two-bit switching. If the pick-off sensors are not properly aligned, if the manufacturing tolerances were low, if environmental effects have resulted in irregularities in the sector matrix, or if there is a significant increase in vibration level due to the run-out or improper maintenance, then it might happen that bit switching from one reading to the next will not take place simultaneously. This will result in ambiguity in data reading during the transition period.

In order to avoid misinterpretation on the data reading, the sectors in the disk are redisplaced in such a way that only one bit is changed from one reading to the next. This is called a *gray code* and is shown in Figure 5.98b. Nevertheless, for an absolute encoder, the gray code is not absolutely essential as the sequence of arrays in the matrix is known beforehand. When there is a data switch, it is possible to check it against the two valid possibilities (or a single one, if the direction of rotation is known).

5.5.10 Other Sensors

It would be impossible, in a single chapter, to describe in detail all the existing sensors, technologies, and applications. This section serves only to give a flavor of the topic. More in depth information can be found (e.g., Ref. [18]) or even by searching for sensor suppliers and manufacturers over the Internet.

Sensors that we missed include (but are not limited to) potentiometers, effort sensors, torque sensors, tactile sensors, gyroscopic sensors, fiber optics, microphones, acoustic emission, chemical sensors, camera-based video systems, etc.

Several areas were identified in which new developments and innovations are made in sensor technology. These areas include the following [18]:

1. Micro- and nanominiature (MEMS) sensors
2. Intelligent sensors with built-in processing capabilities for decision making
3. Embedded and distributed sensor networks, for example, built-in arrays of fiber-Bragg gratings in composite structures [37]
4. Hierarchical sensory architectures, in which low-level sensory information is processed to match higher level requirements

5.6 Logarithmic Scales

5.6.1 Decibel

The decibel (dB) is a logarithmic scale that is used in many fields from electronics to vibration and acoustics. It is used to better represent the ratio between amplitude and a reference value in the y-axis that would not be as conveniently expressed as when using linear units.

The decibel owes its name to Alexander Graham Bell, the founding father of telecommunications. As the name suggests, the decibel is a 10th of a bel. One bel represents the ratio between two power quantities of 10:1.

For example, voltage, current, sound pressure, and velocity are quantities in which square is proportional to power. These quantities are called *root-power* quantities. On the other hand, sound intensity, luminous intensity, and energy density are quantities that are directly proportional to power. These quantities are called *power* quantities [38].

An example of a piece of equipment that delivers results in dB is the sound-level meter (Figure 5.99). Basically, it consists of a spectrum analyzer that uses a microphone as sensor.

5.6.1.1 Power Quantities

Power quantities (also known as *intensity quantities*) are directly proportional to power. The dB level of the ratio between an intensity I and a reference value I_0 is given by

$$L_I = 10\log_{10}\left(\frac{I}{I_0}\right) \tag{5.115}$$

Rearranging this formula, one can determine the intensity magnitude from

$$I = 10^{\frac{L_I}{10}} I_0 \tag{5.116}$$

FIGURE 5.99
A Casella digital sound-level meter is used to measure the sound pressure level from an acoustic source inside an anechoic chamber.

The reference value is the value for which the dB is zero. For example, in acoustics, the reference value for the acoustic intensity is $I_0 \cong 10^{-12}$ W/m^2 This value corresponds to what is considered to be the hearing threshold for an average healthy human adult. When a unit is expressed in dB, it should ideally be represented with its value accompanied by the reference value and its units, for example, 1 dB re 10^{-12} W/m^2, although on many occasions this is omitted. When the reference value is not shown, it is assumed that it is unitary.

To better understand how operations using the dB scale work, let us make a sum between equal power quantities. As an example, let us assume we are making the following sum:

$$70\text{ dB} + 70\text{ dB} = ?$$

We cannot sum dBs as we do with quantities that are expressed on a linear scale. In other words, to say the result is 140 dB would be a wrong answer and quite far from the correct one. First, we must determine the value of the intensity I from Equation 5.116, which, in this case, is

$$I = 10^{\frac{70}{10}} I_0 = 10^7 I_0$$

Now that we have the value of I, we can add them together, obtaining

$$I_{\text{sum}} = I + I = 10^7 I_0 + 10^7 I_0 = 2 \times 10^7 I_0$$

Replacing this value in Equation 5.115, one obtains

$$L_I = 10 \log_{10}\left(\frac{2 \times 10^7 I_0}{I_0}\right) \cong 73\text{ dB}$$

In conclusion, when adding two equal 70 dB power quantities, one obtains

$$70\text{ dB} + 70\text{ dB} \cong 73\text{ dB}$$

Thus, when adding power or intensity quantities, it is easy to prove that the double of the intensity when expressed in decibel is going to be 3 dB larger. One interesting example is the sum of 0 dB + 0 dB, which adds up to 3 dB for power quantities. This happens because 0 dB corresponds to the reference value. For example, in acoustics, the acoustic intensity at 0 dB is 10^{-12} W/m^2, which means that sound already exists. 0 dB does not mean that no sound is being produced; it is the threshold for human hearing. The sound intensity level of absolute silent would be $L_I \to -\infty$ dB.

Similarly, a change in the power ratio by a power of 10 is a 10 dB change.

5.6.1.2 Root-Power Quantities

Root-power quantities (also known as *amplitude quantities*), such as voltage or sound pressure, are quantities for which square is proportional to power. So, it is usual to consider the ratio between the square of the amplitude A and the square of a reference amplitude A_0. The dB level of the ratio between A^2 and a reference value A_0^2 is given by

$$L_{\text{RP}} = 10\log_{10}\left(\frac{A^2}{A_0^2}\right) = 20\log_{10}\left(\frac{A}{A_0}\right) \tag{5.117}$$

Rearranging this formula, one can determine the value for the amplitude magnitude from

$$A = 10^{\frac{L_{\text{RP}}}{20}} A_0 \tag{5.118}$$

Let us now make a sum between equal root-power quantities. As an example, let us assume we are making the following sum:

$$70\text{ dB} + 70\text{ dB} = ?$$

First, we must determine the value of the amplitude A from Equation 5.118, which, in this case, is

$$A = 10^{\frac{70}{20}} A_0 = 3162.3A_0$$

Now that we have the value of A, we can add them together. But because these are root-power quantities, the sum must be done the following way:

$$A_{\text{sum}} = A + A = 3162.3A_0 + 3162.3A_0 = 6324.6A_0$$

Replacing this value in Equation 5.117, one obtains

$$L_{\text{RP}} = 20\log_{10}\left(\frac{6324.6A_0}{A_0}\right) \cong 76\text{ dB}$$

In conclusion, when adding two equal 70 dB root-power quantities, one obtains

$$70\text{ dB} + 70\text{ dB} \cong 76\text{ dB}$$

Thus, when adding root-power or amplitude quantities, it is easy to prove that the double of the amplitude when expressed in decibels is going to be 6 dB larger. One interesting example is the sum of 0 dB + 0 dB, which adds up to 6 dB for root-power quantities.

Similarly, a change in the root-power ratio by a power of 10 is a 20 dB change.

5.6.1.3 Linear versus Logarithmic Frequency Plots

The representation of the magnitude of a function versus frequency is often done in dB. For example, let us assume the frequency spectrum of the RMS value of the acceleration of a faulty impeller is plotted. When a linear scale is used, the spectrum will look like the one shown in Figure 5.100a. This representation highlights the larger peaks in the spectrum. When a logarithmic scale is used (usually expressed in a decibel scale), the spectrum will look like the one in Figure 5.100b. The dB scale is useful to visualize both the smaller and larger amplitudes in the same plot. It is clear in this example that there may be some important peaks between 50 and 150 Hz that are hardly visible using a linear scale whereas using a dB scale they become quite evident. As a disadvantage, the dB scale "amplifies" noise, making it harder to determine whether a peak is just from noise or if it actually is a pattern on the signal.

5.6.1.4 dB Reference Values

Examples of typical reference values for quantities that are represented in a dB scale are shown in Table 5.8 along with their symbols. However, other reference values may be found, and it is not uncommon to find that a unit value (i.e., 1) is simply used. Reference levels have an impact on the dB signal: When the measured signal is larger than the reference value, the dB level is positive; when the measured signal is smaller than the reference value, the dB level is negative.

5.6.1.5 Comparison between the Power and Root-Power dB Scales

A comparison between the dB power and root-power ratio scales is shown in Table 5.9.

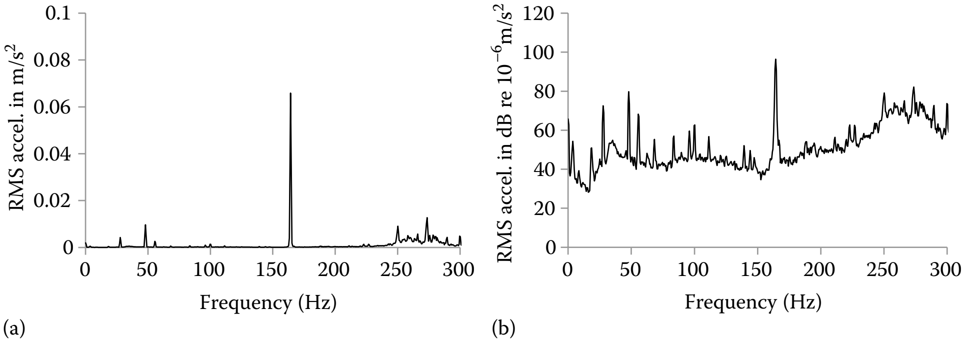


FIGURE 5.100
RMS acceleration spectrum obtained on a impeller: (a) linear scale and (b) decibel (logarithmic) scale.

TABLE 5.8

Typical Reference Values for Quantities That Are Represented on a dB Scale

Quantity Name	Type	Reference Value	Units	Symbol
Sound intensity level	Intensity	10^{-12}	W/m^2	dB SIL
Sound pressure level	Amplitude	20×10^{-6}	Pa	dB SPL
Acceleration	Amplitude	10^{-6}	ms^{-2}	dB re 10^{-6} ms^{-2}
Acceleration	Amplitude	1	ms^{-2}	dB ms^{-2}
Velocity	Amplitude	10^{-9}	ms^{-1}	dB re 10^{-9} ms^{-1}
Velocity	Amplitude	1	ms^{-1}	dB ms^{-1}
Displacement	Amplitude	10^{-12}	m	dB re 10^{-12} m
Displacement	Amplitude	1	m	dB m
Power	Intensity	1	W	dBW
Power	Intensity	1×10^{-3}	W	dBm
Voltage	Amplitude	1	V	dBV
Voltage	Amplitude	1×10^{-3}	V	dBmV
Voltage	Amplitude	1×10^{-6}	V	dBuV
Voltage unloaded (audio)	Amplitude	0.7746	V	dBu
Generic quantity	–	1	unit	dB unit

TABLE 5.9

dB Power and Root-Power Ratio Scales between 120 and –120 dB

dB Level	Power Quantity (Intensity)	Root-Power Quantity (Amplitude)
120	1000000000000 (10^{12})	1000000 (10^{6})
100	10000000000 (10^{10})	100000 (10^{5})
80	100000000 (10^{8})	10000 (10^{4})
60	1000000 (10^{6})	1000 (10^{3})
40	10000 (10^{4})	100 (10^{2})
20	100 (10^{2})	**10**
10	**10**	3.162
6	3.981	**1.995 (~2)**
3	**1.995 (~2)**	1.413
0	1	1
–3	**0.501 (~1/2)**	0.7079
–6	0.251	**0.501 (~1/2)**
–10	**0.1 (10^{-1})**	0.3162
–20	0.01 (10^{-2})	**0.1 (10^{-1})**
–40	0.0001 (10^{-4})	0.01 (10^{-2})
–60	0.000001 (10^{-6})	0.001 (10^{-3})
–80	0.00000001 (10^{-8})	0.0001 (10^{-4})
–100	0.0000000001 (10^{-10})	0.00001 (10^{-5})
–120	0.000000000001 (10^{-12})	0.000001 (10^{-6})

5.6.2 Octave

The octave is a representation of the frequency x-axis widely used in acoustics. One octave is defined as the 2:1 ratio between two frequencies. For example, in music, the middle C (C4) has a frequency of 261 Hz. This means that the upper C (C5), which is said to be "one octave above," will have a frequency of 522 Hz. This octave bandwidth—the difference between the upper and lower frequencies—is 261 Hz. On the other hand, the lower C (C3), which is said to be "one octave below," will have a frequency of 131 Hz. In this case, the octave bandwidth is 131 Hz. The scale is not linear because the difference in frequency between the C5 and C4 is twice as much as the difference in frequency between the C4 and C3.

The octave can also be used in many other applications besides acoustics, for example, in filter design. Historically, bandwidth analysis appeared in any frequency analysis application because it was not computationally practical to use the DFT. *Constant percentage bandwidth filters* can be used instead of the Fourier transform. These consist of analogue or digital passband filters applied directly over the time signal. However, instead of getting spectral lines, one gets bands, which are frequency intervals.

Figure 5.101 shows the bandwidth of one octave, where f_1 is the lower frequency edge, f_2 is the upper frequency edge, and f_c is the central frequency. The relationships between these quantities are given by

$$\frac{f_2}{f_1} = 2^n \tag{5.119}$$

$$f_c = \sqrt{f_2 \cdot f_1} \tag{5.120}$$

$$\%B = \frac{f_2 - f_1}{f_c} \times 100\% \tag{5.121}$$

where n is the number of octaves, and %B is the percentage of bandwidth. Typically, the number of octaves is a fraction: one octave, one third octave, or 1/12 octave although there are many other possible fractions.

As an example, let us assume we want to determine the high-frequency edge of a band seven octaves wide. Equation 5.119 can be used to determine it immediately:

$$\frac{f_2}{10} = 2^7 \Rightarrow f_2 = 1280\ \text{Hz}$$

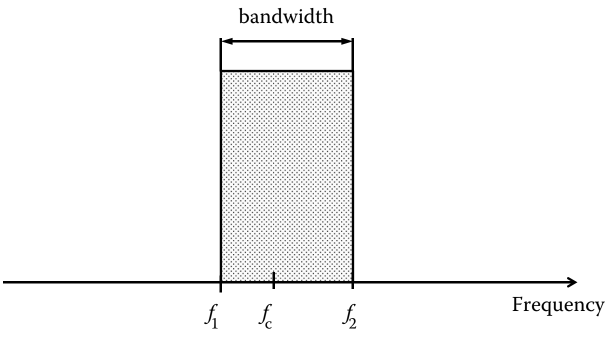


FIGURE 5.101
Octave bandwidth, lower frequency edge, upper frequency edge, and central frequency.

This same result is illustrated in Figure 5.102 for a better understanding. In this picture, it can be seen that each octave's upper frequency is twice as much as the lower frequency and that a total of seven bands are depicted.

Figure 5.102 also shows each octave's bandwidth, which increases with frequency. However, when the percentage of bandwidth is determined from Equations 5.120 and 5.121, the result is found to be constant. One octave has a bandwidth equal to 70.7% of the center frequency, one third octave has a bandwidth equal to 23.2% of the center frequency, and 1/12 octave has a bandwidth equal to 5.78% of the center frequency.

For fractional octaves, the analysis gets a bit more complicated. Let us assume we want to determine the frequencies of the upper edges in one octave bandwidth if 125 Hz is the lower frequency of a one-third octave band. As should now be obvious, in a one octave band with a lower frequency edge at 125 Hz, the upper frequency edge must be 250 Hz. To determine the intermediate frequencies, one uses Equation 5.119 consecutively:

$$\frac{f_2}{125} = 2^{\frac{1}{3}} \Rightarrow f_2 = 157.5 \text{ Hz}$$

$$\frac{f_2}{157.5} = 2^{\frac{1}{3}} \Rightarrow f_2 = 198.4 \text{ Hz}$$

$$\frac{f_2}{198.4} = 2^{\frac{1}{3}} \Rightarrow f_2 = 250 \text{ Hz}$$

This same result is illustrated in Figure 5.103 for better understanding.

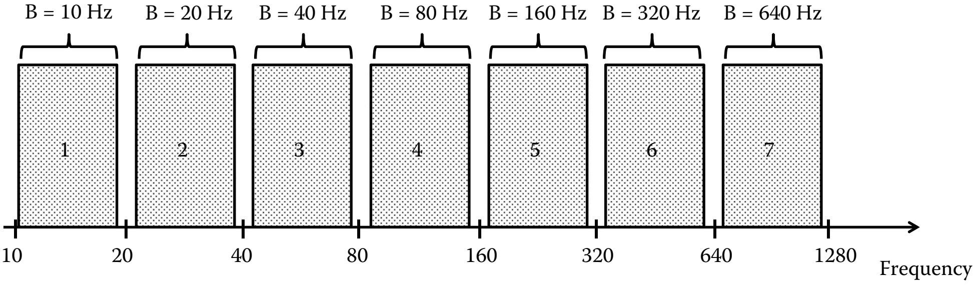


FIGURE 5.102
Lower and upper frequency edges of a band seven octaves wide, starting at 10 Hz.

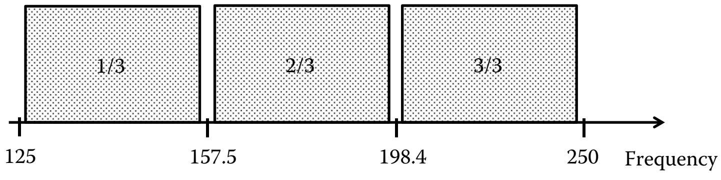


FIGURE 5.103
Lower and upper frequency edges of one-third octave bands in one octave bandwidth, starting at 125 Hz.

5.7 Conclusions

The fundamentals of sensors and signal processing with an emphasis on mechatronics applications were presented in this chapter. It is expected that, after reading this chapter, the engineer has access to the necessary tools that will enable him or her to implement data acquisition systems avoiding common pitfalls in signal processing. Furthermore, a framework has been set so that it serves as a stepping-stone for further study and research to the more interested reader.

There are many other sensors and signal processing techniques besides those introduced in the previous sections. It is impossible to describe them all, mainly due to space limitations. The ones selected were considered to be those that, in addition to being widely used in mechatronics applications, would be capable of illustrating most of the up-to-date techniques used today. For example, the piezoelectric effect, not being exactly novel, is used in a variety of sensing techniques, in the measurement of force, pressure, or acceleration, or to generate ultrasonic bursts. In another application that will be discussed in the sections to follow, piezoelectric elements are used to harvest energy, a topic that has deserved a growing interest in recent years.

References

1. Acoustics, National Physical Laboratory (NPL), Available at: http://www.npl.co.uk/educate-explore/factsheets/acoustics/.
2. Giancoli, D. C., *Physics: Principles with Applications*, Prentice-Hall, Upper Saddle River, New Jersey, USA, 1998.
3. Priemer, R., *Introductory Signal Processing (Advanced Series in Electric and Computer Engineering—Vol. 6)*, World Scientific Publishing, Singapore, 1991.
4. Priestley, M. B., *Non-linear and Non-stationary Time Series Analysis*, Academic Press, London, UK, 1988.
5. Bissel, C. C. and Chapman, D. A., *Digital Signal Transmission*, Cambridge University Press, Cambridge, UK, 1992.
6. Maia, N. M. M, Silva, J. M. M. et al., *Theoretical and Experimental Modal Analysis*, Research Studies Press, Taunton, Somerset, UK, 1997.
7. Fourier, J. B. J., *Théorie Analytique de la Chaleur*, Chez Firmin Didot, père et fils, Paris, France, 1822.
8. Ewins, D. J., *Modal Testing: Theory and Practice*, Research Studies Press, Letchworth, Hertfordshire, UK, 1984.
9. Newland, D. E., *An Introduction to Random Vibrations and Spectral Analysis*, Longman, New York, USA, 1984.
10. Shin, K. and Hammond, J., *Fundamentals of Signal Processing for Sound and Vibration Engineers*, John Wiley & Sons, Chichester, West Sussex, UK, 2008.
11. NI 9234, Available at: http://sine.ni.com/nips/cds/view/p/lang/en/nid/208802.
12. Cooley, J. W. and Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, pp. 297–301, 1965.
13. Bergland, G. D., "A Guided Tour of the Fast Fourier Transform," *IEEE Spectrum*, Vol. 6, pp. 41–52, 1969.
14. Smith, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, California, USA, 1997.

15. Hammond, J. K., "Introduction to Signal Processing—Part I: Fundamentals of Signal Processing," *J. M. Silva and N. M. M. Maia (eds.), Modal Analysis and Testing*, Kluwer Academic Publishers, Dordrecht, Netherlands, NATO Science Series E: Applied Sciences, Vol. 387, pp. 35–52, 1999.
16. LabVIEW™ 2012 Help, National Instruments Co., 2012, Available at: http://zone.ni.com/reference/en-XX/help/371361J-01/.
17. Oppenheim, A. V. and Schafer, R. W., *Discrete-time Signal Processing*, Englewood Cliffs, New Jersey, Prentice Hall, 1989.
18. De Silva, C. W., *Sensors and Actuators: Engineering System Instrumentation*, 2nd Edition, Taylor & Francis, CRC Press, Boca Raton, FL, 2015.
19. Electronic Circuits and Circuit Design Information, Available at: http://www.radio-electronics.com/info/circuits/.
20. Butterworth, S., "On the Theory of Filter Amplifiers," *Experimental Wireless and the Wireless Engineer,* Vol. 7, pp. 536–541, 1930.
21. Matlab R2014a documentation, Curve Fitting Toolbox, Available at: http://www.mathworks.co.uk/help/curvefit/smoothing-data.html.
22. Savitzky, A. and Golay, M. J. E., "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Analytical Chemistry,* Vol. 36, pp. 1627–1639, 1964.
23. Signal Smoothing Algorithms, Available at: http://www.chem.uoa.gr/applets/appletsmooth/appl_smooth2.html.
24. Luo, K., Ying, K., and Bai, J., "Savitzky-Golay Smoothing and Differentiation Filter for Even Number Data," *Signal Processing*, Vol. 85, pp. 1429–1434, 2005.
25. Piezoelectric cubic charge accelerometer 4501A, Brüel & Kjær, Available at: http://www.bksv.com/Products/transducers/vibration/accelerometers/accelerometers/4501A.
26. Montalvão, D., *Determination of Rotational Terms of the Dynamic Response by means of Modal Analysis Techniques*, M.Sc. Thesis, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, 2003.
27. Montalvão, D., Ribeiro, A. M. R., Maia, N. M. M., and Silva, J. M. M., "Estimation of the Rotational Terms of the Dynamic Response Matrix," *Shock and Vibration*, Vol. 11, pp. 333–350, 2004.
28. Bauer, M., Ritter, F., and Siegmund, G., "High-precision Laser Vibrometers based on Digital Doppler-signal Processing," *Proceedings of the 5th International Conference on Vibration Measurements by Laser Techniques: Advances and Applications*, Ancona, Italy, pp. 50–61, 2002.
29. Montalvão, D., *A Modal-based Contribution to the Damage Location in Laminated Composite Plates*, Ph.D. Thesis, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, 2010.
30. Montalvão, D., Ribeiro, A. M. R., and Maia, N. M. M., "Experimental Assessment of a Modal-based Multi-parameter Method for Locating Damage in Composite Laminates," *Experimental Mechanics*, Vol. 51, pp. 1473–1488, 2011.
31. How LVDTs work, Available at: http://www.lvdt.co.uk/how-lvdts-work/.
32. Lage, Y., Reis, L., Montalvão, D., Ribeiro, A. M. R., and Freitas, M., "Automation in Strain and Temperature Control on VHCF with an Ultrasonic Testing Facility," *Journal of ASTM International: Selected Technical Papers of the ASTM 6th Symposium on Automation of Fatigue and Fracture Testing,* in press, 2014.
33. Jones, R. M., *Mechanics of Composite Materials*, Taylor & Francis, Philadelphia, Pennsylvania, USA, 1999.
34. Montalvão, D. and Fontul, M., "Harmonica: Stepped-Sine Spectrum Analyser for Transfer Function Measurement and Non-Linear Experimental Assessment," *Proceedings of M2D'2006—5th International Conference on Mechanics and Materials in Design*, paper no. A0519.0506, Porto, Portugal, 2006.
35. Scott, P., *Experimental Investigation into a Novel Design Concept for a Modular PEMFC Stack*, Ph.D. Thesis, School of Engineering and Technology, University of Hertfordshire, 2013.

36. Hot-Wire and Hot-Film Anemometers, Thermopedia, Available at: http://www.thermopedia.com/content/853/.
37. Montalvão, D., Ribeiro, A. M. R., and Maia, N. M. M., "A Review on Vibration Based Structural Health Monitoring with Special Emphasis on Composite Materials," *Shock and Vibration Digest*, Vol. 38, pp. 295–324, 2006.
38. IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, 7th Edition, Standards Information Network, IEEE Press, New York, USA, 2000.

6

Bio-MEMS Sensors and Actuators

Farbod Khoshnoud, Clarence W. de Silva, and Ibrahim Esat

CONTENTS

6.1 Introduction 222
6.2 Bio-MEMS Actuators 222
    6.2.1 Artificial Muscles 222
        6.2.1.1 Ionic EAP 223
        6.2.1.2 Dielectric EAP 225
    6.2.2 Ciliary Actuators 226
    6.2.3 Nanotweezers for Micromanipulation of Biomolecules 229
    6.2.4 Application of Capillary Valves in Microfluidic Devices 233
    6.2.5 Drug Delivery 235
    6.2.6 Biomolecular Systems 240
6.3 Bio-MEMS Sensors 240
    6.3.1 Triglyceride Biosensor 241
    6.3.2 Bio-MEMS Sensor for C-Reactive Protein Detection 241
    6.3.3 Glucose Detection 242
    6.3.4 MEMS Force Sensor for Protein Delivery 243
    6.3.5 Tissue Softness Characterization 243
    6.3.6 Blood Cell Counter 244
    6.3.7 Acoustic Sensor 245
6.4 Conclusions 245
References 245

SUMMARY This chapter presents an introduction to micro-electromechanical systems (MEMS), which includes microminiature sensors and actuators with biomedical applications. In the section on actuators, artificial muscle, nanotweezers for micromanipulations, drug delivery systems, biomimetic ciliary actuators, and so on are discussed. In the section on sensors, several examples are presented, including triglyceride sensing, protein detection, glucose detection, force sensing for protein delivery, tissue softness characterization, blood cell counting, and acoustic sensing. Sensor and actuator systems are realized using various technologies, including electrostatic, electromagnetic, piezoelectric, and thermo-pneumatic principles, and the use of multifunctional materials, such as electroactive, bimetallic, and shape memory alloys.

6.1 Introduction

Mechatronic systems are instrumented typically with sensors, actuators, and related hardware [1–5]. Sensors are used for a variety of purposes, including monitoring, fault prediction, detection and diagnosis, experimental modeling, and control. In particular, sensors are necessary to measure excitations (inputs) and responses (outputs) of a mechatronic system. Actuators facilitate the system to produce the desired output and behave in a desired manner, according to performance specifications [1]. Sensor and actuator systems can be realized through various technologies, including electrostatic, electromagnetic, piezoelectric, and thermo-pneumatic, and by the use of multifunctional materials such as bimetallic, shape memory alloy (SMA), and ionic conductive polymer film (ICPF).

Micro-electromechanical systems (MEMS) use microminiature sensors and actuators. MEMS technologies allow higher accuracy and speed and lower power consumption and cost [6–8]. The MEMS fabrication techniques are normally based on semiconductor (e.g., silicon) device fabrication procedures, which include deposition of material layers, patterning by photolithography and etching to produce the required shapes. MEMS devices for biomedical applications, referred to as biomedical MEMS (bio-MEMS), are devices developed for improving the health associated with various physiological and medical applications. Several examples of such MEMS sensors and actuators are presented in this chapter.

6.2 Bio-MEMS Actuators

Bio-MEMS actuators, such as artificial muscle, nanotweezers for micromanipulations, drug delivery systems, and biomimetic ciliary actuators are discussed in this section. The MEMS actuator devices realized by various actuation mechanisms, such as electrostatic, electromagnetic, piezoelectric, and thermo-pneumatic, and the use of multifunctional materials, such as electroactive, bimetallic, and shape memory alloys, are presented in this section.

6.2.1 Artificial Muscles

In this section, some of the technologies in developing artificial muscles using electroactive polymers (EAP) are discussed. EAP are particularly attractive for developing biologically inspired mechanisms, devices, and robots, such as artificial muscles, to mimic the movements of humans, animals, and insects [9–12]. Polymers are lightweight, easily fabricated in various geometry configurations, and low cost, which makes them attractive as actuators. The parameters that are considered in designing an actuator include energy density, strain, actuation pressure, response time, environmental tolerance, fabrication complexity, reliability, and efficiency. An actuator is developed or selected for a specific application, depending on the design requirements. An actuator normally cannot perform ideal functionality in all conditions. For instance, electromagnetic actuators can perform well as artificial muscles on a macro scale. However EAP can provide higher efficiency relative to electromagnetic actuators at small scales [13]. Polymer actuators may be realized as electrostrictive polymers [14–18], piezoelectric polymers [19], shape memory polymers [20], electrochemically actuated conducting polymers [21–26], polymer-based air-gap electrostatic devices [27], etc.

Many EAP materials exist, and they can be divided into ionic (involving mobility or diffusion of ions) and electronic, such as dielectric EAP (driven by electric field or Maxwell forces) [9,10]. Dielectric EAP actuators (DEAs) are capable of exhibiting large strain with actuation pressure per unit mass in the order of 103 Pa.m^3/kg while consuming low energy [28,29]. Dielectric polymers have high mechanical energy density and can be operated in air. However, dielectric polymers require very high activation fields (>10 V/µm) [11]. Compared to ionic EAP such as IPMC, the activation of ionic polymers requires only 1–2 volts. However, ionic EAP need to maintain wetness if they are not particularly designed as self-contained encapsulated activators for the use in dry environments [30]. Moreover electromechanical coupling and compatibility of ionic polymers is limited (although they are ideal for biomimetic devices).

6.2.1.1 Ionic EAP

Ionic polymeric-conductor composites (IPCCs) are suitable candidates for biomimetic sensors, robotic actuators (Figure 6.1) and artificial muscles [31–36] in the range of micro to macro scales. Ionic polymers in a composite form with a conductive medium, such as metals, for example, ionic polymeric-metal composites (IPMCs), or other conductive materials can provide relatively large deflections (for instance, compared to piezoelectric material) when they are exposed to an electric field. Conversely, the material generates electrical energy in response to dynamic loads.

The actuation/sensing mechanism of the IPMC can be described as in Figure 6.2 [31,36]. Under the electric field $E(x, y, z, t)$, the ions (i.e., Na$^+$) move within the material toward the cathode and carry with them water molecules. When the water molecules accumulate near the cathode, a local overpressure is created, which consequently tends to deform the material.

The potential due to the electric field can be expressed by Poisson's equation as

$$\nabla^2 \varnothing(x,y,z,t) = -\frac{\rho(x,y,z,t)}{\varepsilon} \tag{6.1}$$

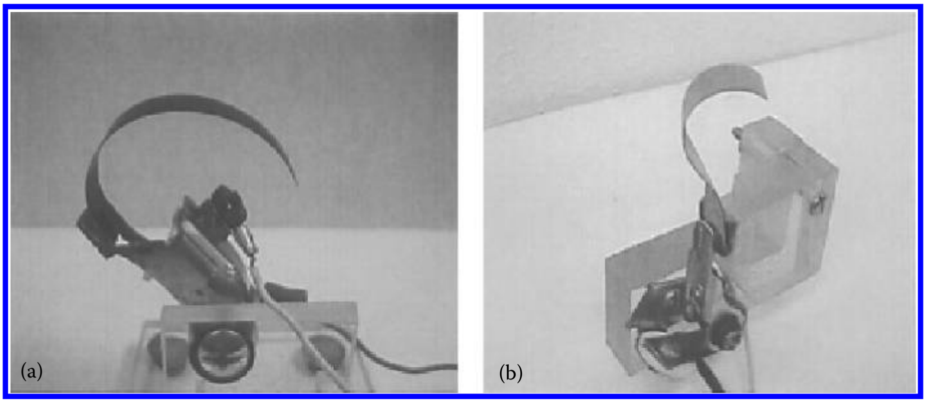


FIGURE 6.1
Ionic polymeric-metal composites (IPMCs). (a) Large bending deflection of an IPMC strip. (b) Combined bending and torsional deflection. (From Smart Materials and Artificial Muscles Laboratory, University of Maine. With permission.)

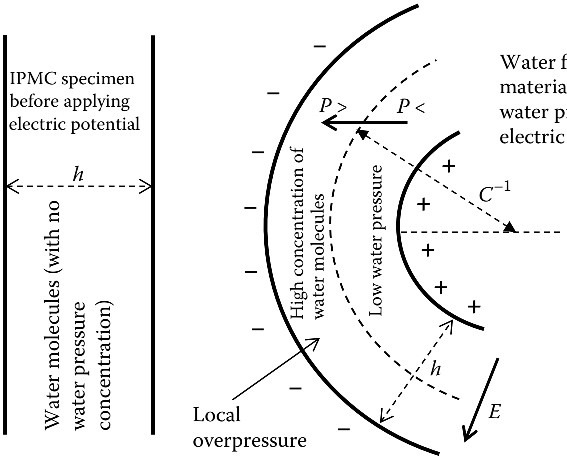

IPMC specimen before applying electric potential
h
Water molecules (with no water pressure concentration)
P >
P <
High concentration of water molecules
Low water pressure
C^{-1}
Water flux within the IPMC material structure due to the water pressure when under electric field, E, is applied
Local overpressure
h
E
−
+

FIGURE 6.2
Actuation/sensing mechanism of the ionic polymeric-metal composites (IPMCs).

The electric charge transport, J, normal to the membrane, and the solvent transport (water flux), Q, can be written in terms of the electric field, E, and the mechanical load due to water pressure, p, as [31,36]

$$
\begin{aligned}
J(x,y,z,t) &= \sigma E(x,y,z,t) - L_{12}\nabla p(x,y,z,t) \\
Q(x,y,z,t) &= L_{21}E(x,y,z,t) - K\nabla p(x,y,z,t)
\end{aligned}
\tag{6.2}
$$

where σ is the membrane conductance, K is the Darcy permeability, and $L_{12} = L_{21} = L$ are cross coefficients. For electrodes that are impermeable to water, $Q = 0$, the pressure created by the electric field is

$$
\nabla p(x,y,z,t) = \frac{L}{K}E(x,y,z,t) \tag{6.3}
$$

where the pressure gradient ∇p is responsible for the deflection of the material with a curvature of C. When a mechanical load is applied to the material, although no electric energy is generated and consumed, that is, the electric charge transport, $J = 0$, then

$$
E(x,y,z,t) = \frac{L}{\sigma}\nabla p(x,y,z,t) \tag{6.4}
$$

The energy per unit width of a strip of a IPMC due to the applied torque of T can be written in terms of the curvature, C, the length of the strip, s, and the angular deflection, θ, as follows:

$$
U = \frac{1}{2}BC^2s + T\theta \tag{6.5}
$$

where the bending constant, B, is given in terms of Young's modulus, Y [37], the strip thickness, h, and Poission's ratio, ν, by

$$B = \frac{Yh^3}{12(1-\nu^2)} \tag{6.6}$$

For an active IPMC strip under a field, E, when $T = 0$, the resulting stresses due to the water pressure, p, is in the order of $E\varepsilon$ as

$$p(x, y, z, t) \cong Y\varepsilon \tag{6.7}$$

where ε is the deformation of the strip at a distance z from the midplane of the strip in the order of $\varepsilon = -Cz$.

The pressure gradient can be calculated by

$$\nabla p(x,y,z,t) = \frac{\partial p(x,y,z,t)}{\partial z} = -kYC \tag{6.8}$$

where the coefficient k is given by [36]

$$k^{-1} = (1 + \nu)(1 - 2\nu) \tag{6.9}$$

When the field and torque acting simultaneously on the strip, the deflection contribution due to the torque can be added to the deflection contribution due to the field resulting in the following total curvature.

$$-C = \frac{\nabla p(x,y,z,t)}{kY} + \frac{T}{B} \tag{6.10}$$

As an example of performance of a IPMC, when a step voltage of 1.5 volts and current of 157 mA is applied to a semidry IPMC cantilever beam of 20 mm × 5 mm × 0.2 mm dimensions, the displacement response of the free end of the cantilever is about 12 mm [31,36].

6.2.1.2 Dielectric EAP

The working principle of dielectric electroactive actuators is analogues to capacitive actuators. Elastomeric polymer film sandwiched between two compliant electrodes is actuated by electrostatic force [10,38] (Figure 6.3).

When voltage is applied, the compliant electrodes apply pressure on the polymer film in between the electrodes due to electrostatic pressure and squeeze the polymer [10,13,38]. The effective actuation pressure can be expressed by

$$P = \varepsilon\varepsilon_0 E^2, E = \frac{V}{z} \tag{6.11}$$

where E denotes the electric field, ε is the dielectric constant of the polymer, ε_0 is the vacuum permittivity, z is the thickness of the elastomer, and V is the applied voltage.

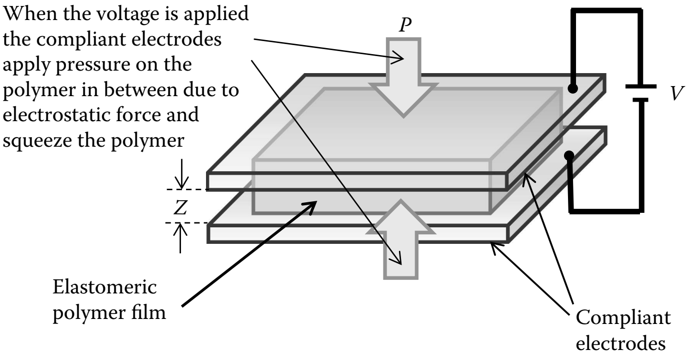

When the voltage is applied the compliant electrodes apply pressure on the polymer in between due to electrostatic force and squeeze the polymer
P
V
Z
Elastomeric polymer film
Compliant electrodes

FIGURE 6.3
Working principle of dielectric electroactive actuators.

For small strains of the polymer, ϵ, and free boundary condition, we have [13]

$$\epsilon = -\frac{P}{Y} = -\frac{\varepsilon\varepsilon_0 E^2}{Y} \tag{6.12}$$

The relative permittivity, known as the dielectric constant, for EAP is in the range of two to 12. As a comparison, silicon as a dielectric material can also provide high relative permittivity of 11.68. However due to the relatively lower value of polymer Young's modulus, which makes it soft, unlike silicon, the dielectric EAP can generate larger deflections.

6.2.2 Ciliary Actuators

Cilia are hair-like organelles growing out of the surface of most body cells, which can provide locomotion to liquids along internal tissue in animals. Figure 6.4 shows cilia in a scanning electron microscope image of lung trachea epithelium [39]. Cilia can be found with dimensions ranging from 7 μm to 50 μm in lengths and about 250 nm in diameter.

High-aspect-ratio micro-/nanostructures are favorable in designing MEMS as actuation arm mechanisms. Biomimetic cilia devices realized by the advancements in MEMS technology allow the development of cilia MEMS devices for actuation and manipulation at the micro-scale [40].

Various types of cilia-like actuators have been proposed using polyimide bimorph ciliary array [41–44], electrostatically actuated polymer artificial ciliary actuator [45,46], and magnetically actuated polydimethylsiloxane (PDMS) nanorods [47] for manipulating objects at submillimeter scales and generating forces in the range of tens of nanonewtons.

Polyimides are known for excellent mechanical properties, good thermal stability, and chemical resistance. Polyimides compounded with graphite or glass fiber reinforcements have flexural strengths of up to 340 MPa and flexural moduli of 21,000 MPa [48], which are maintained to temperatures of up to 452°C and are also inherently resistant to flame combustion. A bimorph biomimetic cilia microactuator is designed using two layers of polyimide with different coefficients of thermal expansion (CTE) [41–44], in which aluminium plates, interconnects, and resistive heaters are embedded in between the two layers of polyimide for electrostatic and thermal actuation, respectively. Figure 6.5 presents a schematic representation of a single (1 mm × 1 mm) cilium. An actuator mechanism can be designed based on a scenario that the cantilever is initially curved as in Figure 6.5b. The

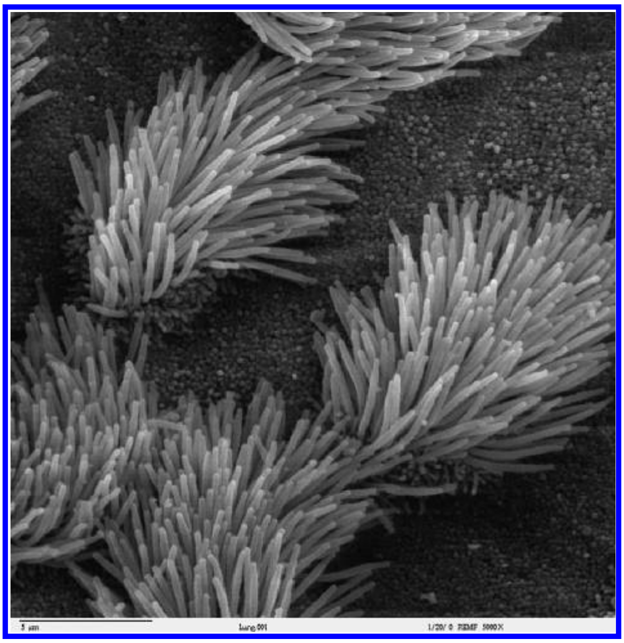

5 µm
Lung 001
1/20/0 REMF 5000X

FIGURE 6.4
Cilia in scanning electron microscope image of lung trachea epithelium. (From Electron Microscope Facility, Dartmouth Medical School, Hanover, NH. With permission.)

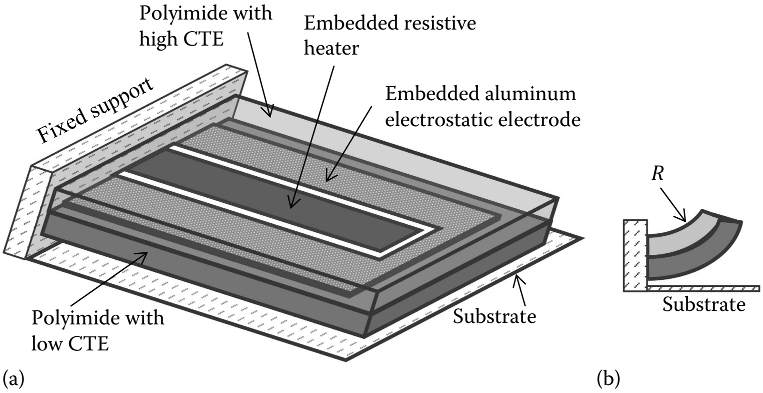

Polyimide with high CTE
Embedded resistive heater
Fixed support
Embedded aluminum electrostatic electrode
R
Polyimide with low CTE
Substrate
Substrate
(a)
(b)

FIGURE 6.5
A single bimorph biomimetic cilia microactuator consists of two layers of polyimide with different thermal expansion coefficients, a resistive heater, and an electrostatic plate, (a) when heated, (b) before heat is applied.

actuator is then flattened when the high CTE polyimide expands with a larger deformation relative to the low CTE polyimide.

The passage of electric current through the resistive heater (Figure 6.5) releases heat. This heat results in thermal expansion of the polyimide layers with different CTE, therefore bending the material with the radius of curvature R given by [41]

$$R = \frac{h}{\epsilon_h - \epsilon_l} \tag{6.13}$$

where h is the thickness of the actuator, and ϵ_h and ϵ_l are the strains in the polyimide layers with high and low CTE, respectively. The required force for deflecting the actuator tip can be estimated by

$$F = \frac{3(EI)_{\text{equiv.}}\,\delta}{L^3} \tag{6.14}$$

where $(EI)_{\text{equiv.}}$ denotes the equivalent flexural rigidity of the actuator with all the elements, including the polyamide, aluminum, and the heater layers; L is the length of the actuator; and δ is the displacement of the cantilever tip. An example of the actuator parameters and testing results are given below:

$L = 430$ μm, $\delta = 114$ μm, $R = 800$ μm, the resistance of the resistive heater = 1500 Ω with a dissipation of 16.7 mW when a potential of 5 V is applied, the capacitance = 300 fF for an area of 24,400 μm^2, force $F = 76$ μN [41].

Generally, thermal actuators are known to have a high demand for power, and therefore, an electrostatic actuation mechanism combined with the thermal actuation can be designed to lower the overall power consumption. Moreover, the electrostatic scheme can be adopted as a sensing mechanism.

A layer of aluminum electrostatic electrodes is embedded within the cantilever length (Figure 6.5). The capacitance of the electrostatic capacitor is created by a stack of several dielectric layers and an air gap. The capacitance of the actuator can be approximated by

$$C = \epsilon_r \epsilon_0 \frac{A}{d} \tag{6.15}$$

where ϵ_r is the relative dielectric constant for the air gap, ϵ_0 is the electric constant of vacuum ($\epsilon_0 \approx 8.854 \times 10^{-12}$ F/m), A is the area of the aluminium plate, and d is the mean distance of the curved cantilever relative to a substrate. As an example for a capacitance of 300 fF per cilium, applying a voltage of 500 V can produce a deflection of 15 μm, which is lower than the deflection produced by thermal actuation. A combined thermal and electrostatic actuation can help in reducing the power consumption by first applying the thermal actuation to flatten the actuators and placing them near the substrate and then applying the electrostatic force to hold the actuators in the flattened position. Such micro-biomimetic cilia actuators can be employed in developing general-purpose micromanipulation tools

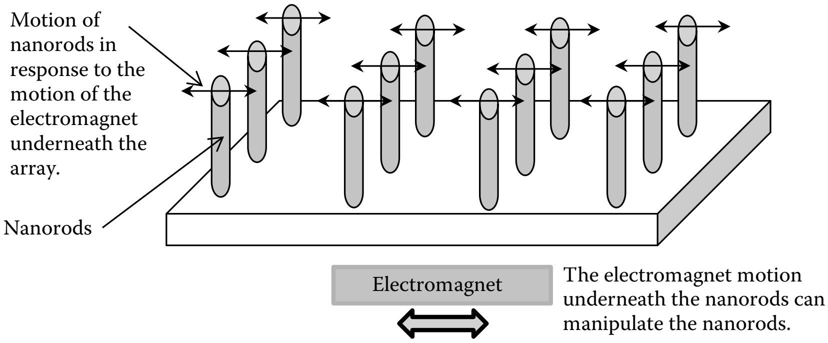

Motion of nanorods in response to the motion of the electromagnet underneath the array.
Nanorods
Electromagnet
The electromagnet motion underneath the nanorods can manipulate the nanorods.

FIGURE 6.6
Actuation of PDMS nanorods using an electromagnet.

for small objects [41], docking small spacecraft [43], and omnidirectional mobile microrobots [44].

PDMS nanorods have been fabricated with high aspect ratios of 125 with diameters ranging from 200 nm to 1 μm [47]. Nanorods contain iron oxide particles and therefore respond to magnetic fields. The actuation and manipulation of the nanorods is achieved by an electromagnet moving underneath the nanorod arrays (see Figure 6.6).

6.2.3 Nanotweezers for Micromanipulation of Biomolecules

Atomic force microscopes (AFM), optical tweezers (OT), and magnetic tweezers (MT) are the most commonly used biomechanical instruments for single-molecule micromanipulation [49–59]. Information about stretching displacement of a molecule under an applied force helps in understanding the underlying mechanisms [55], such as DNA conformation [56], chromatin organization [57], and biomolecular interaction dynamics [58,59]. The double helix conformation of DNA molecules exhibit enhanced stiffness, which allows regulating biomolecular interactions and is vital for life.

An AFM consists of a flexible cantilever with a sharp tip (Figure 6.7). The free end of a molecule can be picked up by the sharp tip of the AFM cantilever. Once the molecule is trapped between the base substrate and the tip, the molecule is stretched by displacing the substrate. The force acting on the AFM tip, due to the DNA tension force, bends the cantilever. The bending of the cantilever can be monitored by detecting the deviation of a laser beam due to the deflection of the cantilever using a photodiode detector (Figure 6.7). By knowing the equivalent spring stiffness of the cantilever and the displacement of the cantilever due to bending, the force applied to the molecule is obtained as the product of the stiffness and the displacement.

For a 1.5-μm contour length DNA, 500 pN force extends the molecule to 4 μm before breakage [60]. The molecule breakage also depends on the stretching speed, which is usually in the range of 0.15 μm/s to 3 μm/s [61].

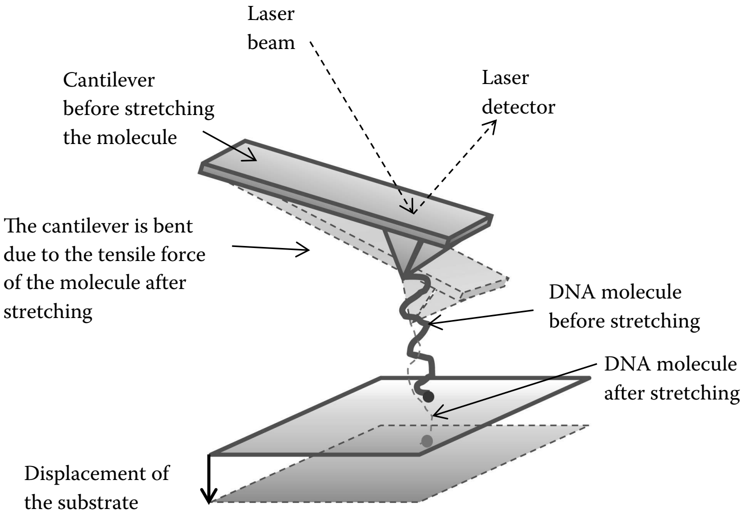

Laser beam
Cantilever before stretching the molecule
Laser detector
The cantilever is bent due to the tensile force of the molecule after stretching
DNA molecule before stretching
DNA molecule after stretching
Displacement of the substrate

FIGURE 6.7
Molecular manipulation by an atomic force microscope (AFM).

OT are the most commonly used tools for stretching and manipulating biomolecules in liquid [53]. When a laser beam is focused with an objective lens, a dielectric particle in the vicinity of the focused beam is attracted to the focal point due to the transfer of momentum from the scattering of incident photos [62,63]. Figure 6.8 illustrates the working concept of OT. A DNA molecule is captured in between a dielectric bead, which is pulled toward the optical trap due to the attractive force of the laser focal point and a bead held by a micropipette.

If a dielectric bead is held at the focal point and then gradually pulled away through displacement x, an attractive force F tends to return the bead to the optical trap. An equivalent stiffness k can be obtained from this practice where $F = k.x$, assuming a linear force–displacement relationship. Stiffness k is used as a calibration constant in the experiment of determining the force required for stretching a molecule. This force can be calculated by multiplying the k value by the displacement of the center of the bead from the optical trap (Figure 6.8), where this displacement is measured experimentally. The same linear force–displacement relationship is used in calculating the force. Image analysis techniques are capable of locating a micrometer-sized bead within a 10 nm accuracy [53]. This gives a force resolution of 0.5 pN for a trap stiffness of 50 μN/m. The magnitude of the forces that can be generated using OT technique is up to 100 pN. In biomechanical experiments, molecules are prepared with a bead bound on each end of the molecule [64,65].

In MT, the molecule is prepared by one end bound to a glass coverslip and the other end to a paramagnetic bead [56]. A magnetic force applied by (electro)magnets to the paramagnetic bead can stretch a molecule in MT technique. The magnitude of the applied force can be tuned by varying the separation distance between the bead and the magnet or electrically tuned by varying the current in an electromagnet [66].

When the magnetic force is applied to the paramagnetic bead, it stretches the molecule while exhibiting fluctuations in the motion of the bead and the molecule. Therefore in order to determine the force using MT, the Brownian motion of the bead is recorded, and the force is then calculated based on the recorded fluctuations. The vertical magnetic force, F (Figure 6.9), can be expressed by $F = kTL/\langle\delta x\rangle^2$, where kT is the thermal energy, L is the extension of the molecule, and δx is the Brownian fluctuations [56,66]. $\langle\delta x\rangle$ is determined from a recorded video of the fluctuations of the bead displacement and image processing.

Forces can be measured in the range of femtonewtons to 100 pN using MT. In order to obtain sufficient recorded samples of the Brownian fluctuations, typically the measurement is carried out for the duration of 1 min for a 10 pN force and over 1 hr for determining a force in the range of 10 fN [53].

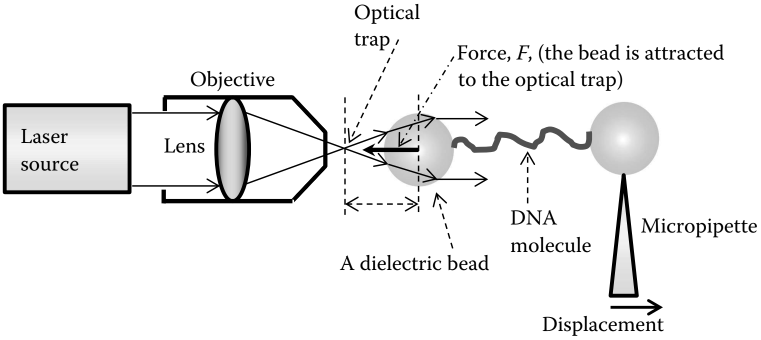


FIGURE 6.8
Optical tweezers (OT).

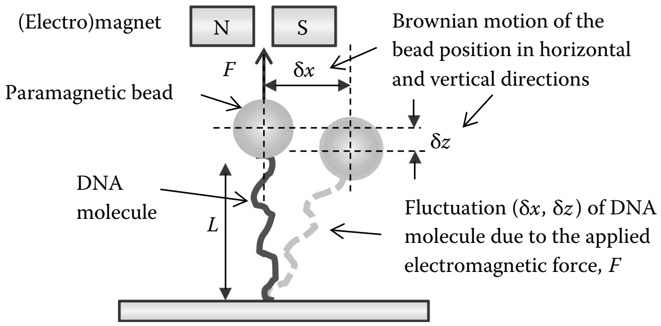

(Electro)magnet
N
S
F
δx
Brownian motion of the bead position in horizontal and vertical directions
Paramagnetic bead
δz
DNA molecule
L
Fluctuation (δx, δz) of DNA molecule due to the applied electromagnetic force, F

FIGURE 6.9
Magnetic tweezers (MT).

Micro-nano-electromechanical systems (MNEMS) tweezers are composed of opposing sharp tips for capturing molecules, an actuator for stretching molecules, and a sensor to measure displacements [53,67–71]. Trapping DNA boundless with (MNEMS) tweezers is carried out by bringing the tweezer tips in contact with a droplet of DNA solution on a coverslip [67]. When a high AC electric field of 1 MHz is applied between the tweezers arms, the DNA molecules are stretched and attracted to the tweezer tips. In order to trap a single DNA, molecular isolation is performed in a microfluidic device prior to the trapping [72]. A schematic representation of silicon-based MNEMS tweezers and the working principle are illustrated in Figure 6.10.

Figure 6.11 [68–71] shows a silicon nanotweezer device and a trapped DNA bundle by the tweezers.

Both sensing and actuation of a tweezer arm is carried out using an electrostatic comb scheme. The electrical signal generated from the movement of the comb arm is measured

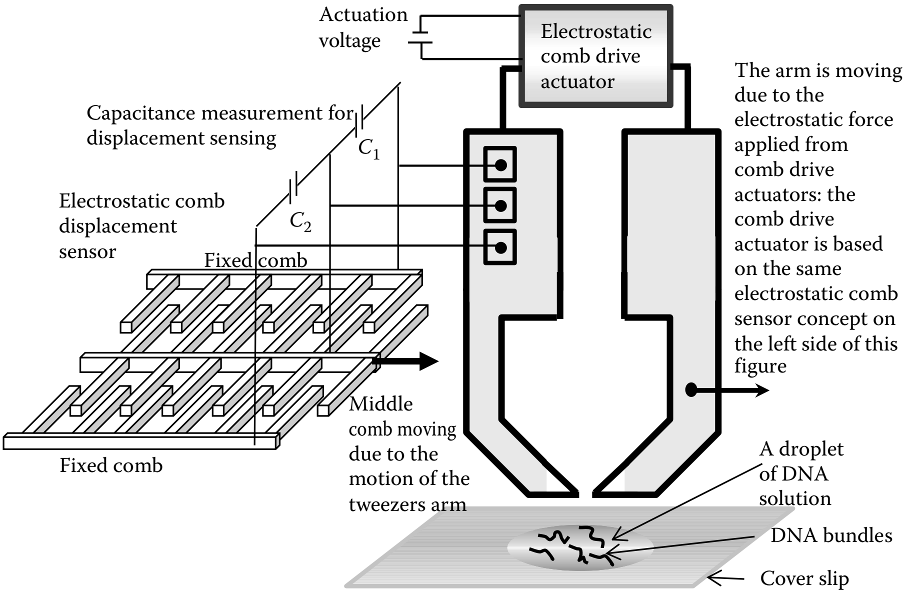

Actuation voltage
Electrostatic comb drive actuator
The arm is moving due to the electrostatic force applied from comb drive actuators: the comb drive actuator is based on the same electrostatic comb sensor concept on the left side of this figure
Capacitance measurement for displacement sensing
C_1
Electrostatic comb displacement sensor
C_2
Fixed comb
Fixed comb
Middle comb moving due to the motion of the tweezers arm
A droplet of DNA solution
DNA bundles
Cover slip

FIGURE 6.10
Working principle of micro-nano-electromechanical systems (MNEMS) tweezers.

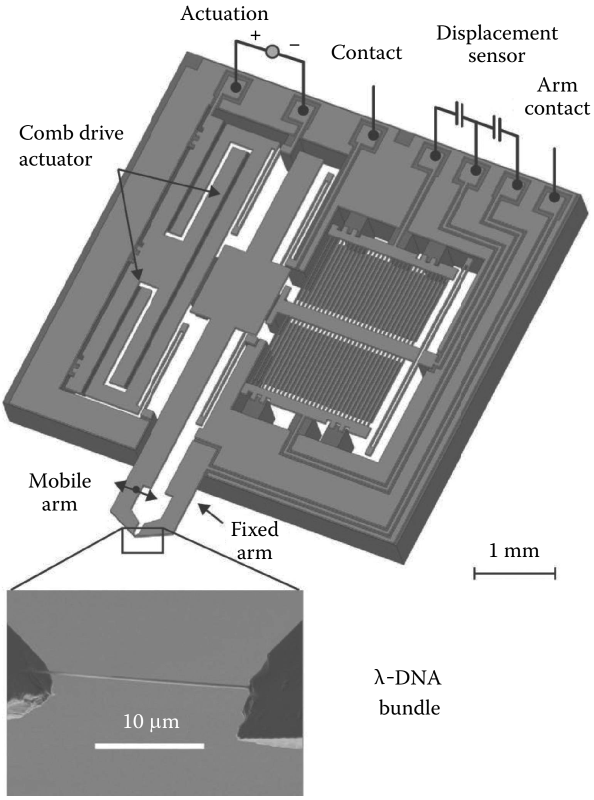


FIGURE 6.11
Micro-nano-electromechanical systems (MNEMS) tweezers device, and a stretched DNA bundle. (From LIMMS/CNRS-IIS, Center for International Research on MicroMechatronics CIRMM Institute of Industrial Science, University of Tokyo. With permission.)

by monitoring the capacitance variation ΔC. It is translated into the displacement of the tweezers Δx using the following relationship [53]:

$$\Delta C \approx -2n\varepsilon_0 Lh\left(\frac{1}{d_1^2}-\frac{1}{d_2^2}\right)\Delta x \text{ and } \Delta C = C_1 - C_2 \tag{6.16}$$

where n denotes the number of comb arms, ε_0 is the permittivity of air, L is the projected length of the overlapped moving and fixed comb arms, h is the thickness of the combs, and d_1 and d_2 are the initial distance of the moving comb relative to the fixed comb for C_1 and C_2, respectively. As an example, the overall sensitivity of the differential capacitive sensor may be in the range of $V_{out}/\Delta x = -3.29$ mV/μm, where V_{out} is the output voltage of the sensor. Precise actuation and displacement sensing provide molecular extension control in the range of 0.3 nm resolution and the force resolution of 10 nN (0.3 nm × 35 N/m), where the global mechanical stiffness of the tweezers is ~35 N/m [53].

Manipulating microscale objects for biomedical applications requires fine resolution movement, high reliability, and intuitive master interface and can be further improved by force sensing for haptic feedback. Figure 6.12 illustrates the concept of a multiscale manipulation with which a desired displacement input is given to the system at the

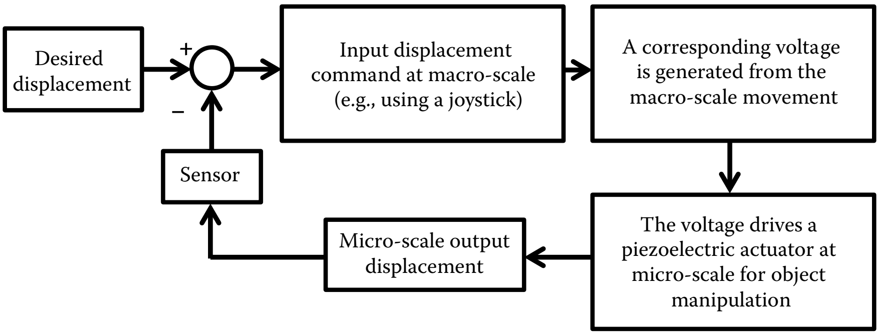

Desired displacement
+
−
Input displacement command at macro-scale (e.g., using a joystick)
A corresponding voltage is generated from the macro-scale movement
Sensor
The voltage drives a piezoelectric actuator at micro-scale for object manipulation
Micro-scale output displacement

FIGURE 6.12
A multiscale actuation mechanism for converting a macroscale motion into a microscale displacement.

macroscale (e.g., using a joystick). Then the macroscale motion is calibrated and mapped for microscale manipulation with which a corresponding voltage is generated to drive a piezoelectric actuator at microscale. The actuator motion can be fine-tuned using a feedback control system. Bilateral micromanipulation is increasingly being applied in intracellular injection in which it eliminates the limitations of manual manipulation. Fine manual manipulations can only be achieved after long and lengthy training of an operator and usually provide a low success rate due to relying on visual information coming from microscopy. Biological cells exhibit irregular shapes, and they are very soft, which makes them very vulnerable in manual manipulation. In a bilateral manipulation system, a master interface allows steering of a slave manipulator that is equipped with triaxial force sensors for haptic force feedback [73]. The triaxial force sensing device can measure forces in the range of 0–3 N for normal forces and ±50 mN for tangential forces with a resolution of 11 bits. The actuation of the manipulator at the microscale is carried out when master interface information is mapped to the micromanipulator by means of analog voltage applied to a piezoelectric actuator. The feeling of an object being manipulated is obtained by applying a known force to an object and measuring the object deformation, which gives a corresponding spring stiffness associated with the force-displacement values. The system is calibrated in term of the calculated stiffness, so it can give measurement information and furthermore provide sensing for haptic feedback and feeling the object that is being manipulated. This equivalent spring stiffness gives the property of the object (e.g., cell). Then this information is fed back to the manipulation system for motion control purposes.

6.2.4 Application of Capillary Valves in Microfluidic Devices

Cells can adapt to changes in their natural environment. Cells can sense and convert mechanical forces into chemical signals via mechanotransduction. Regulations of various biological processes, such as gene expression, adhesion, migration, and cell fate, which are essential to the maintenance of tissue homeostasis, are involved with the applied mechanical forces in the biological environment [74]. Mechanical stimulation *in vitro* (multidimensional strain) can selectively differentiate mesenchymal progenitor cells into the ligament lineage [75]. Stem cells are able to detect and respond to alterations in the stiffness of their surrounding environment via induction of lineage-specific differentiation. Mechanobiology studies the control of stem cell biology through mechanical factors [76]. Mechanobiology is defined as the study of mechanisms by which cells detect and respond to mechanical stimuli. External mechanical forces applied to stem cells are transduced through a variety of actuation schemes, which allow differentiation processes.

Combination of fluidics with a microactuator is considered to be a promising tool in mechanobiology studies of single cells [77–80] as an alternative to tedious instrumentation currently used by biophysicists, such as optical tweezers, micropipetting, and AFM indentation. Handling fluid at the micro-device clearances can be achieved by making the surfaces at the clearances hydrophobic or using capillary action [80–83].

The influence of mechanical stress (e.g., stretching and compressing) on cell differentiation can be examined using an electrostatically actuated silicon chip and an open fluidic system that incorporates a passive capillary valve for partial horizontal immersion of the device in the culture medium. The principle of a push–pull electrostatic actuator based on capillary action is depicted in Figure 6.13 [80]. The electrostatic force, $F(t)$, due to the capacitance between the electrostatic comb drive moves a silicon bar horizontally in a microfluidic chamber. The microfluidic chamber contains the culture medium. The cells in the medium attach to the moving and stationary silicon bars. Force is applied to the cells that are attached in between moving and stationary silicon bars due to the motion of the moving bar. The electrostatic force is generated by an AC voltage, $V(t)$. The electrostatic force can be represented by

$$F(t) = F_0 \sin(\omega t), \text{ where } F_0 = 2\frac{\partial C}{\partial x} V_0 V_\mathrm{L} \tag{6.17}$$

where ω denotes the frequency of the AC signal, t is the time, F_0 is the amplitude of the applied force that depends on the amplitude of the DC voltage, V_L, the amplitude of the AC voltage, V_0, and the derivative of the comb-drive capacitance with respect to displacement.

The equation of motion of the comb drive can be written as

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \frac{F_0}{M}\sin(\omega t) \tag{6.18}$$

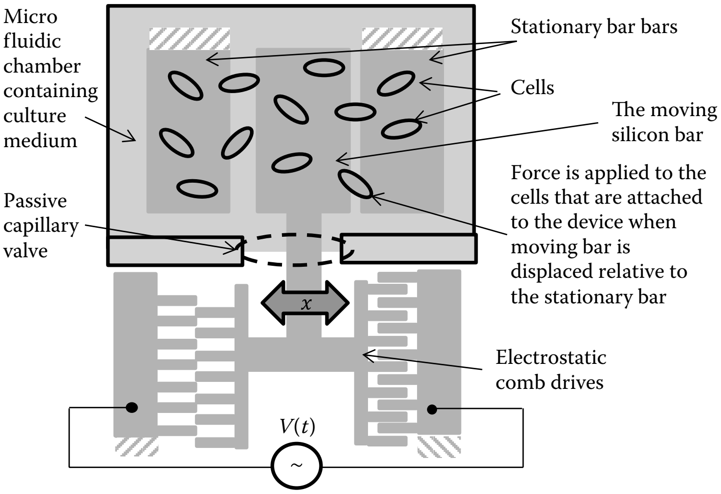

Micro fluidic chamber containing culture medium
Passive capillary valve
Stationary bar bars
Cells
The moving silicon bar
Force is applied to the cells that are attached to the device when moving bar is displaced relative to the stationary bar
x
Electrostatic comb drives
V(t)
~

FIGURE 6.13
Stretching and compressing cells.

where x is the displacement of the comb drive as in Figure 6.13, ζ is the damping ratio of the fluid inside the micro chamber, ω_n is the natural frequency of the device, ω is the excitation frequency due to the applied AC voltage, and M is the effective mass of the moving structure.

It should be noted that the performance of this device is strictly dependent on adhesion of cells to the moving and stationary silicon. This device is feasible for mechanobiology actuation purposes only if the cell adhesion is done in a controlled manner. The electrostatic comb drive mechanism presented above is a general scheme commonly used in MEMS sensing and actuation.

One of the characteristics of this device is the utilization of capillary action, which can be very attractive in microfluidic MEMS devices. The passive capillary valve is presented in Figure 6.13. The capillary valve allows leak-free insertion of the MEMS device silicon bar into the microfluidic chamber. When the length scales are smaller than capillary length, λ_c, the silicon bar can enter the microfluidic opening without any fluid leaking. λ_c can be given by $\lambda_c = \sqrt{\frac{\gamma}{\rho g}}$ [81], where γ is the surface tension of the liquid–air interface ($\gamma = 0.07$ J/m^2 for water–air interface at 37°C), ρ is the liquid density, and g denotes gravity. Capillary length for water is $\lambda_c \approx 2.7$ mm.

For a microtube with a circular cross-section of diameter $2a$, we have

$$\Delta P_{\text{surf}} = \frac{2\gamma}{R} = \rho g h, \text{ and } R = \frac{a}{\cos\theta} \tag{6.19}$$

where ΔP_{surf} is the Young-Laplace pressure drop along the liquid–air interface curve, R denotes curvature radius of the meniscus, θ is the contact angle at the liquid–solid–air interface, and h is the equilibrium height of the liquid column.

6.2.5 Drug Delivery

Drugs or chemical agents should be delivered with a specific degree of concentration so that the desired treatment effect is achieved [84]. The rate of drug delivery or the target area of the drug in conventional dosage forms, such as oral delivery and injection, cannot be easily controlled and are often associated with an immediate or rapid drug release. This can lead to high initial concentration of the drug in the body above the level of toxicity and gradual dissolution in an ineffective period of time [85]. The controlled drug delivery technology has been studied since the 1970s. Microdevices can improve drug therapy because they allow precise and complex dosing [86]. An integrated drug delivery system (DDS) consisting of a drug reservoir, micropumps, valves, microsensors, microchannels, and the necessary related circuits is shown in Figure 6.14 [87]. Micropumps can be categorized into mechanical and nonmechanical. In mechanical micropumps, an actuator performs the pumping function [84]. The actuation in a mechanical pump can be realized by various processes, including electrostatic, piezoelectric, thermo-pneumatic, bimetallic, shape memory alloy (SMA), ionic conductive polymer film (ICPF), electromagnetic, and phase change. Nonmechanical pumps include electrical, chemical, magnetic, and surface-tension-driving micropumps.

Maximum flow rate $\dot{Q}_{\max}$, maximum backpressure $h_{\max}$, pump power P, and pump efficiency η are the parameters used for characterizing a micropump. For incompressible flow

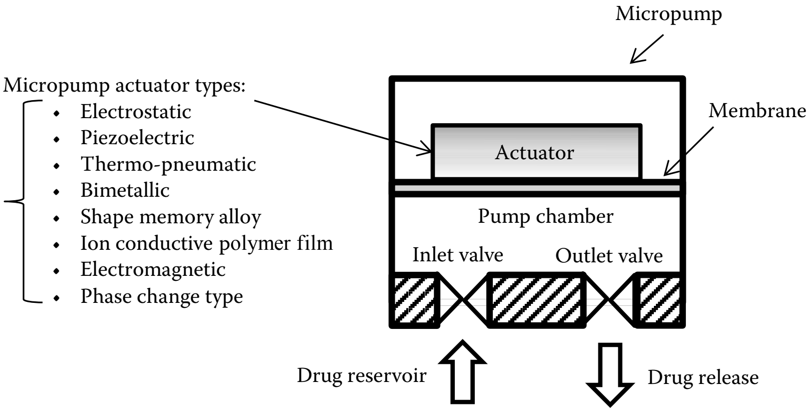

Micropump
Micropump actuator types:
Electrostatic
Piezoelectric
Thermo-pneumatic
Bimetallic
Shape memory alloy
Ion conductive polymer film
Electromagnetic
Phase change type
Membrane
Actuator
Pump chamber
Inlet valve
Outlet valve
Drug reservoir
Drug release

FIGURE 6.14
Schematic representation of a drug delivery system.

and by neglecting viscous work and heat transfer, the pump head h or net head can be given by the steady flow energy equation as

$$h = \left(\frac{p}{\gamma} + \frac{u^2}{2g} + z\right)_{\text{out}} - \left(\frac{p}{\gamma} + \frac{u^2}{2g} + z\right)_{\text{in}} \tag{6.20}$$

where p = pressure, $\gamma = \rho g$ = pressure head, g = acceleration due to gravity, ρ = fluid density, u = fluid velocity, and z = elevation. For a micropump we normally have $u_{\text{out}} \cong u_{\text{in}}$ and $z_{\text{out}} \cong z_{\text{in}}$, which gives the maximum pump head as

$$h_{\max} \approx \frac{p_{\text{out}} - p_{\text{in}}}{\gamma} \tag{6.21}$$

The delivered power by the micropump and the ideal efficiency of the pump can be given by

$$P = \rho g h_{\max} \dot{Q}_{\max}\text{, and } \eta = \frac{P}{P_{\text{actuator}}} \tag{6.22}$$

where P_{actuator} = ideal actuator power in the absence of any energy losses.

The electrostatic actuation mechanism in a micropump is achieved by two opposing electrically charged palates [88–91]. Figure 6.15 gives a schematic representation of an electrostatic drug delivery system.

A controlled voltage applied to the electrostatic actuator manipulates a membrane and generates the periodic fluid pumping action. The electrostatic attraction force F can be represented by $F = \varepsilon A V^2/2x^2$, where ε = dielectric constant of the medium between the opposing plates, A = area of the plates, V = applied voltage, and x = distance between the plates. The advantages of electrostatic micropumps are low power consumption, which is of the order of 1 mW and fast response time [87]. The deflection of the diaphragm can be controlled by the applied voltage. The disadvantages include the small actuation displacement, which is normally limited to 5 μm with the applied voltages of about 200 V.

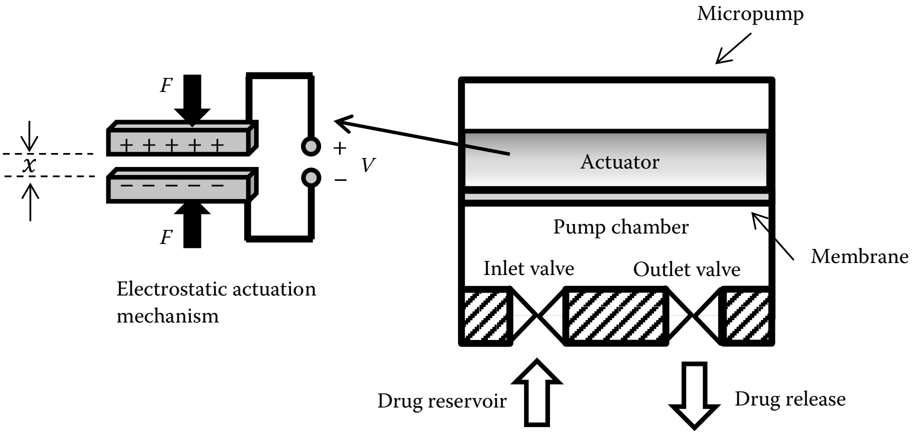

Micropump
F
+ + + + +
− − − − −
x
F
+
V
−
Actuator
Pump chamber
Inlet valve
Outlet valve
Membrane
Electrostatic actuation mechanism
Drug reservoir
Drug release

FIGURE 6.15
Schematic representation of an electrostatic drug delivery system.

In piezoelectric micropumps [92–106] the actuation is generated when a voltage is supplied to a piezoelectric material for driving the pump membrane. Figure 6.16 presents the schematic diagram of a piezoelectric drug delivery system.

Piezoelectric actuators can exhibit a large actuation force with a fast response time. Although the structure of piezoelectric actuators is simple, the fabrication is complex for MEMS applications because the piezoelectric materials cannot be processed easily. A relatively high voltage (e.g., 100–600 V) is required for operating a piezoelectric actuator. These actuators normally give small displacements. For a millimeter-scale piezoelectric pump device, typical characteristic parameters are in the range of up to thousands of microliters per minute for flow rate, tens of kPa for back pressure, less than 10 mWs of power consumption, and an operation frequency of hundreds of Hz [87]. Higher flow rates can be achieved using a piezoelectric actuation mechanism when compared to electrostatic actuation.

In a piezoelectric micropump, when the piezoelectric material is subjected to an external force, it can generate a voltage. The electromechanical relationship for a linear piezoelectric continuum can be written by employing the principle of conservation of energy, and obeys

$$\dot{U} = T_{ij}\dot{S}_{ij} + E_i\dot{D}_i \tag{6.23}$$

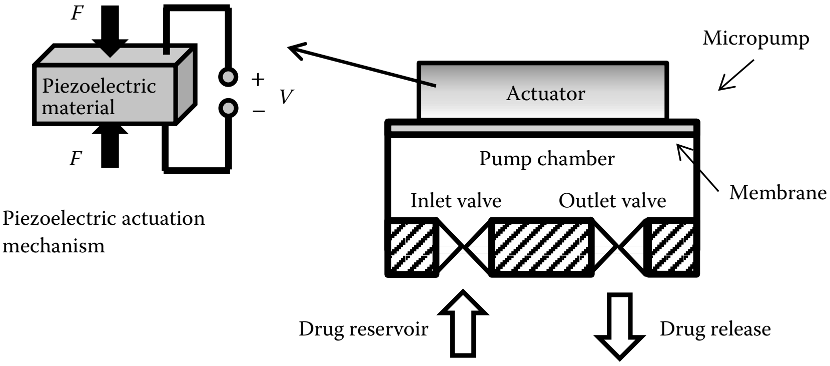

F
Piezoelectric material
F
+
V
−
Micropump
Actuator
Pump chamber
Inlet valve
Outlet valve
Membrane
Piezoelectric actuation mechanism
Drug reservoir
Drug release

FIGURE 6.16
Schematic representation of a piezoelectric drug delivery system.

Here $\dot{U}$ denotes the rate of change of the stored energy density of the piezoelectric continuum with respect to time, T_{ij} is the stress tensor, $\dot{S}_{ij}$ is the rate of the change of the strain tensor with respect to time, E_i is the electrical field tensor, and $\dot{D}_i$ represents the rate of change of the electric displacement tensor with respect to time.

In thermo-pneumatic micropumps [107–114], an air chamber behind the pump chamber membrane is expanded when the air is heated. Heating and cooling the air inside the chamber provides the periodic actuation of the pump (see Figure 6.17).

Heating is achieved using a resistive heating mechanism. As an example, a millimeter-scale pump device can deliver tens of microliters per minute of flow at tens of kPa of pressure and more than 100 mWs of power consumption at a voltage less than 100 V and a frequency lower than 10 Hz. The relatively complicated structure, slow response time, and low efficiency are the disadvantages of this type of pumps.

Shape memory alloy (SMA) material, such as titanium nickel alloy (TiNi), can be used as the actuation mechanism in a micropump [115–118]. Figure 6.18 presents a schematic diagram of an SMA drug delivery pump. SMAs change phase in heating and cooling cycles and deform and restore the original shape in each cycle.

A millimeter-size pump device can deliver a flow of about 340 μl/min at a pressure of 100 kPa. However the power consumption is relatively high, which is in the range of 0.5 W. A bimetallic micropump consists of two attached materials with different coefficients of

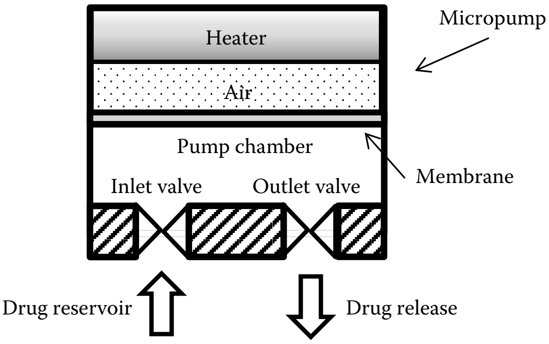

Heater
Air
Micropump
Pump chamber
Membrane
Inlet valve
Outlet valve
Drug reservoir
Drug release

FIGURE 6.17
Schematic representation of a thermo-pneumatic drug delivery system.

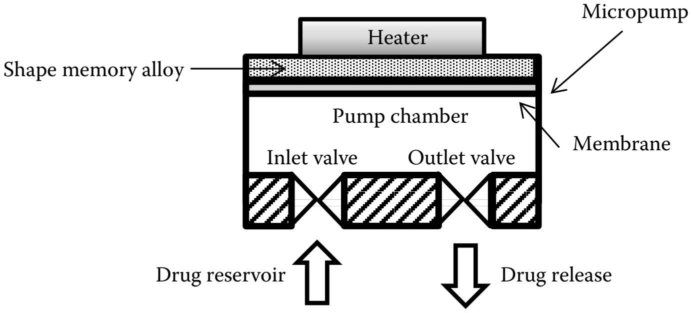

Micropump
Heater
Shape memory alloy
Pump chamber
Membrane
Inlet valve
Outlet valve
Drug reservoir
Drug release

FIGURE 6.18
Schematic representation of a SMA drug delivery system.

thermal expansion. Heating and cooling cycles of the bimetallic material provide the actuation of the micropump [119–121]. Figure 6.19 presents a schematic diagram of a bimetallic micropump.

Bimetallic micropumps can operate with low voltage (e.g., 5.5 V). However, the response time can be slow (e.g., 0.5 Hz). The reported flow rate and maximum backpressure for this type of pump are in the range of 45–336 μl/min and 0.5–12 kPa, respectively [119–121].

An ion conductive polymer film (ICPF) micropump [122–125] operates when a voltage is applied to an ICPF strip (Figure 6.20). ICPF consists of a polymer sandwiched between thin films of material with high electrical conductivity (e.g., gold). The working principle of ICPF is similar to that of the artificial muscle discussed in Section 6.2.1. The flow rate of such a pump can be in the range of approximately 4.5–37.8 μl/min with a driving voltage of 1.5 V [122–125]. Advantages of ICPF include low driving voltage, fast response, and biocompatibility. However it can be difficult to fabricate.

Electromagnetic actuation of a micropump requires soft magnetic elements and a coil. Electromagnetic actuation can provide large force and displacement, unlike the electrostatic mechanism, and has high power consumption (e.g., 0.5 W) and high heat energy loss [95,126–129]. However, microfabrication of magnetic material is limited to a few types of materials. A schematic representation of an electromagnetic actuation micropump is given in Figure 6.21.

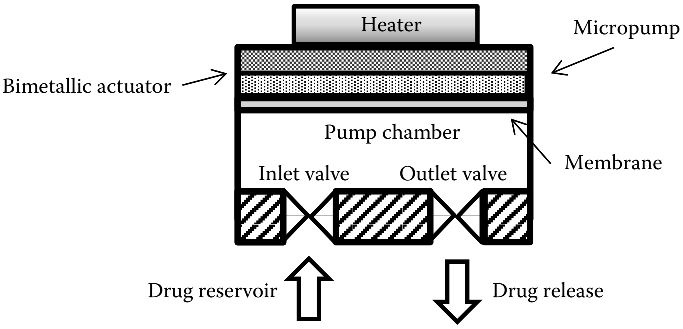

Heater
Micropump
Bimetallic actuator
Pump chamber
Membrane
Inlet valve
Outlet valve
Drug reservoir
Drug release

FIGURE 6.19
Schematic representation of a bimetallic drug delivery system.

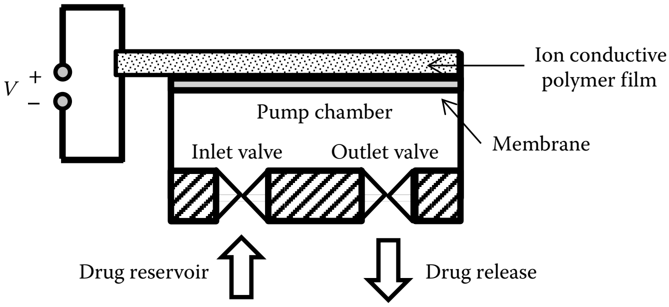

V
+
−
Ion conductive polymer film
Pump chamber
Membrane
Inlet valve
Outlet valve
Drug reservoir
Drug release

FIGURE 6.20
Schematic representation of an ICPF drug delivery system.

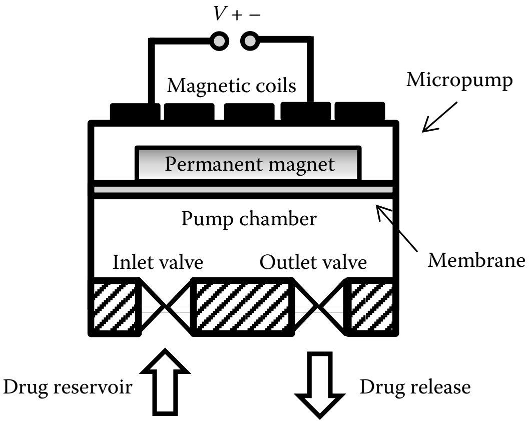


FIGURE 6.21
Schematic representation of an electromagnetic drug delivery system.

6.2.6 Biomolecular Systems

Biological organisms use their sensory systems to detect changes in their environment. The sensing ability includes fold-change detection with a response that depends only on fold changes in the input. Synthetic biology allows the development of biomolecular systems for understanding the architecture and the response of cellular systems to inputs, exploring the system's behavior, and building a theoretical framework to understand the experimental outcome. As an example, a simplified mathematical model for a synthetic circuit [130] is given by

$$\begin{aligned}\dot{x} &= \alpha_1 u - \beta_1 x - kxy + \gamma_1 z \\ \dot{y} &= \alpha_2 u - kxy \\ \dot{z} &= kxy - \gamma_1 z\end{aligned} \tag{6.24}$$

This is a mathematical representation that can be used to explore the phase space of circuit behavior in response to inputs. The parameters used in these equations are given below. A more detailed explanation of these parameters can be found in the references [130–134]. The aim of the present section is to introduce the importance of biomolecular systems. In the mathematical model (Equation 6.24), $u = [\mathrm{A^{tot}}]$, $x = [\mathrm{iMG}]$, $y = [\mathrm{rMG}]$, and $z = [\mathrm{rMG{\cdot}iMG}]$, where rMG (malachite green aptamer) and its inhibitor iMG are the product of two RNA species, A is a single DNA activator, α_1 and α_2 are functions of the k_ps (RNAP concentration), and β_1 and γ_1 are functions of RNase R concentration and secondary structure of RNA molecules [130].

6.3 Bio-MEMS Sensors

Mechatronic systems operate with the use of sensors and actuators [1–5]. Sensors are necessary to measure excitations (inputs) and responses (outputs). Sensors are used for a variety of purposes, including monitoring, fault prediction, detection and diagnosis, experimental modeling, and control. Sensor systems can be realized with various sensing processes, including electrostatic, electromagnetic, piezoelectric, and thermo-pneumatic, and the use of multifunctional materials, such as bimetallic, shape memory alloy (SMA), and ionic conductive polymer film (ICPF). MEMS sensors that use microminiature sensors allow higher accuracy and speed and lower power consumption and fabrication cost [6–8]. MEMS-based biological sensors are applied in physiological, medical, and health applications. Examples of such sensors include triglyceride sensing, protein detection, glucose detection, force sensor for protein delivery, tissue softness characterization, a blood cell counter, and an acoustic sensor.

6.3.1 Triglyceride Biosensor

Composite porous silicon/polysilicon micro-cantilevers can be used in biosensing applications, such as triglyceride sensing. Micro-cantilevers have been employed in detecting cells, proteins, heavy metals, and other chemical and biological species. Arrays of these cantilevers are capable of detecting multiple parameters simultaneously. When bio molecules are adhered to a coated micro-cantilever surface, the cantilever mass increases. Variation in the cantilever mass changes the natural frequencies of the cantilever. This variation is detected by a Doppler vibrometer [135] (Figure 6.22). Porous layers on the surface of the micro-cantilever provide a larger sensing area for the sensor.

6.3.2 Bio-MEMS Sensor for C-Reactive Protein Detection

Protein catalyzes biochemical reactions to transport and store nutrients, provides protection from viruses and bacteria, and transmits biological signals [6]. C-reactive protein (CRP) concentration in human serum is below 1 μg/ml in a healthy body. Increase of CRP in the bloodstream due to an infection can cause cardiovascular disease and heart attacks. Fluorescence-based bio-sensing is a common technique in biological molecule sensing. However, it is costly and requires complicated labeling processes of target molecules with dye. A 200-μm-long micro-cantilever in a wireless MEMS sensor is realized to detect disease

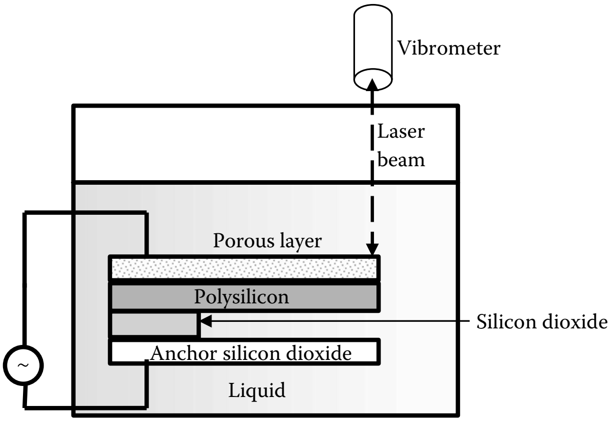


FIGURE 6.22
Biosensing using resonant frequency measurement of a cantilever beam in a solution. (From Khoshnoud, F. and De Silva, C. W., *IEEE Instrumentation and Measurement*, 15, 1, 8–14, 2012; Fernandez, R. E., Stolyarova, S., Chadha, A., Bhattacharya, E. and Yael Nemirovsky, *IEEE Sensors Journal*, 9, 12, 2009.)

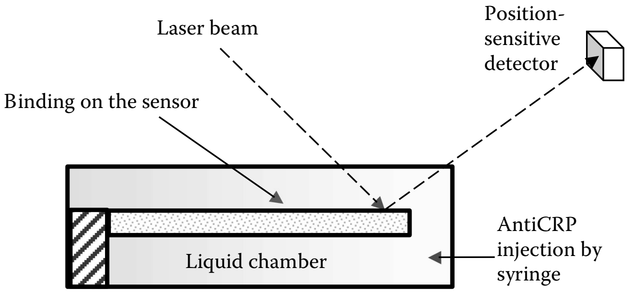

Laser beam
Position-sensitive detector
Binding on the sensor
AntiCRP injection by syringe
Liquid chamber

FIGURE 6.23
Schematic diagram of the C-reactive protein sensor. (From Khoshnoud, F. and De Silva, C. W., *IEEE Instrumentation and Measurement*, 15, 1, 8–14, 2012; Chen, C., Hwang, R., Huang, L., Lin, S., Chen, H., Yang, Y., Lin, Y., Yu, S., Lin, Y., Wang, Y., Chou, N. and Lu, S., *IEEE Transactions on Biomedical Engineering*, 56, 2, 2009.)

related to CRP [136]. When an anti-CRP is injected to the sensor liquid chamber, it adheres to the cantilever surface, and the intermolecular nanomechanical interactions within the bio-linker layer bends the cantilever due to the biomolecular interactions between the CRP and anti-CRP change. The deflection of the micro-cantilever can be detected optically or by using a piezoresistive mechanism (Figure 6.23).

6.3.3 Glucose Detection

Glucose is measured in diabetic patients to monitor blood sugar levels. MEMS sensors can be realized for monitoring glucose levels in a patient's blood [6,137–139]. MEMS-based glucose level measurement techniques include the use of electrochemical, impedance, electrophoretic, thermal, optical, and colorimetric principles. Continuous monitoring of glucose for diabetes management can also be attained by magnetically driven vibrating micro-cantilevers. In this sensor, a micro-cantilever oscillates in a micro-fluidic chamber containing a glucose-sensitive polymer solution (Figure 6.24). The cantilever detects the viscosity change due to affinity binding between glucose and poly acrylamide-ran-3-acrylamidophenylboronic acid (PAA-ran-PAAPBA). The cantilever is paced in a micro-chamber filled with PAA-ran-PAAPBA solution. When glucose permeates through a membrane, it changes the viscosity of the solution. Damping of the vibrating cantilever changes due to the change in the viscosity of the solution. The vibration behavior of the cantilever is monitored optically. Glucose concentration can be measured in the range of 27 mg/dL to 324 mg/dL using such a device.

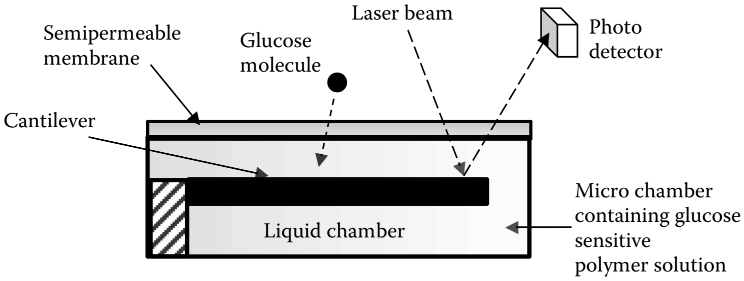

Laser beam
Photo detector
Semipermeable membrane
Glucose molecule
Cantilever
Micro chamber containing glucose sensitive polymer solution
Liquid chamber

FIGURE 6.24
MEMS glucose sensor.

6.3.4 MEMS Force Sensor for Protein Delivery

A MEMS capacitive force sensor as shown in Figure 6.25 [6,140] is capable of sensing forces up to 110 μN with a resolution of 33.2 nN for obtaining mechanical properties of soft hydrogel microcapsules for protein delivery. Hydrogel microcapsules are applied to the controlled delivery of therapeutic agents (e.g., protein drugs) and the encapsulation of living cells. The mechanical strength of hydrogel microcapsules determines if they can survive in the needle tract during injection, in the blood capillaries, or in the applied tissues. Characterization of the mechanical properties of hydrogel microcapsules using MEMS sensors allows the design of efficient drug and cell delivery systems, and avoiding dose dumping, cell death, or immunoresponse.

6.3.5 Tissue Softness Characterization

In open surgery, trauma is caused by incisions. Minimally invasive surgery (MIS) allows smaller incisions, less pain during the recovery period, reduced blood loss and scarring, faster recovery time, shorter hospitalization, fewer complications due to infection, better prognosis, better cosmetic results, and reduced overall cost [141]. The surgeon has to identify the type of tissue (e.g., fatty, muscular, vascular, or nerve) when making an incision into the tissue. Cutting a wrong tissue (e.g., nerve tissues) can cause the loss of motor control in patients. MIS tools can identify the type of tissue by measuring the forces applied to the grasped tissues and the local distribution of the forces to quantify the tissue softness. The force sensor should be capable of detecting and locating the concentrated loads due to the presence of any anatomical feature (e.g., lumps or tubular features, pulsating or nonpulsating features) in the grasped tissue. Identification and classification of different types of tissue during surgery is an important feature in the design of surgical tools.

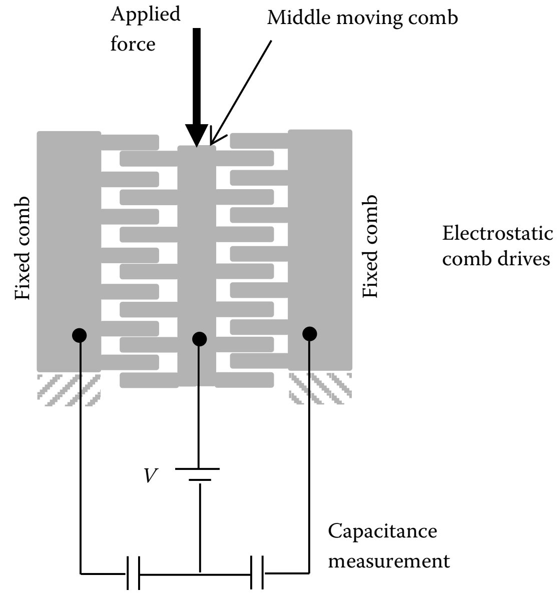


FIGURE 6.25
Schematic diagram of a capacitive force sensor.

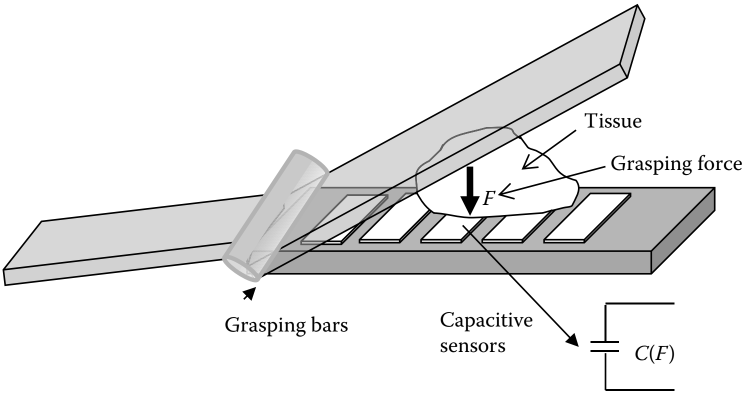


FIGURE 6.26
Grasper for tissue softness sensing.

MEMS tactile sensors integrated with MIS graspers offer tactile sensing capabilities [6]. A MEMS tactile sensor that is fabricated from a uniaxial polyvinylidene fluoride (PVDF) film [141] can measure the magnitude and location of a force and tissue softness. A compressive force applied by graspers (Figure 6.26) is used to measure the softness of tissues. For an applied force, a soft tissue produces a smaller deformation in the sensors, which can be calibrated to identify the softness of the tissue.

6.3.6 Blood Cell Counter

Blood consists of plasma, red blood cells, white blood cells, and platelets. Conventional blood cell counters, based on the aperture-impedance method and a light-scattering method, are bulky, which makes them not suitable for their use at the point of care. MEMS technology can be utilized to overcome this limitation [6]. A MEMS blood cell counter device can detect blood cells [142] in an injected dilute blood sample in an aperture by monitoring the change of the electrical resistance. The electrical resistance changes due to the concentration of the injected diluted blood sample in the aperture, in which the change of resistance is proportional to the volume of the cells. The number of blood cells is counted from the total number of electrical pulses (Figure 6.27), through the change in resistance when a constant voltage is applied across the aperture. Red and white blood cells are distinguished from the difference of the electrical pulse heights.

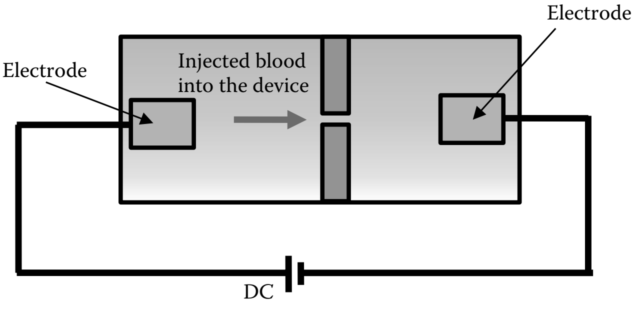


FIGURE 6.27
A blood counter sensor.

6.3.7 Acoustic Sensor

A MEMS acoustic sensor can assist human hearing. It can be developed using piezoresistive, condenser, and piezoelectric methods [6]. Although new hearing devices such as digital hearing aids and cochlear implants are used today, the MEMS acoustic method is still applied in cellphones, micro-personal digital assistants, portable multimedia players, and voice recognition [143].

6.4 Conclusions

The principles, application, and importance of bio-MEMS sensors and actuators were presented in this chapter. Illustrative examples were given. In the category of actuators, artificial muscle, nanotweezers for micromanipulations, drug delivery systems, and biomimetic ciliary actuators were discussed. In the category of sensors, several examples were presented including triglyceride sensing, protein detection, glucose detection, force sensing for protein delivery, tissue softness characterization, blood cell counting and an acoustic sensor.

References

1. De Silva, C. W., *Sensors and Actuators: Engineering System Instrumentation*, 2nd Edition, Taylor & Francis, CRC Press, Boca Raton, FL, 2015.
2. De Silva, C. W., *Mechatronics—An Integrated Approach*, CRC Press/Taylor & Francis, Boca Raton, FL, 2005.
3. De Silva, C. W. (Ed.), *Mechatronic Systems—Devices, Design, Control, Operation, and Monitoring*, Taylor & Francis/CRC Press, Boca Raton, FL, 2007.
4. De Silva, C. W., *Mechatronics: A Foundation Course*, Taylor-Francis, CRC Press, Boca Raton, FL, 2010.
5. De Silva, C. W., *Modeling and Control of Engineering Systems*, CRC Press/Taylor & Francis, Boca Raton, FL, 2009.
6. Khoshnoud, F. and De Silva, C. W., "Recent advances in MEMS sensor technology—Biomedical applications," *IEEE Instrumentation and Measurement*, Vol. 15, Issue 1, pp. 8–14, 2012.
7. Khoshnoud, F. and De Silva, C. W., "Recent advances in MEMS sensor technology—Mechanical applications," *IEEE Instrumentation and Measurement*, Vol. 15, Issue 2, pp. 14–24, 2012.
8. Khoshnoud, F. and De Silva, C. W., "Recent advances in MEMS sensor technology—Thermo-fluid and electrical," *IEEE Instrumentation and Measurement*, Vol. 15, Issue 3, pp. 16–20, 2012.
9. Bar-Cohen, Y., *Electroactive Polymer (EAP) Actuators as Artificial Muscles—Reality, Potential and Challenges*, 2nd Edition, Vol. PM136, SPIE Press, Bellingham, WA, pp. 1–765, 2004.
10. Zhang, Q. M., Huang, C., Xia, F. and Su, J., "Electric EAP, Chapter 4," in Y. Bar-Cohen, (Ed.), *Electroactive Polymer (EAP) Actuators as Artificial Muscles*, 2nd Edition, Vol. PM136, SPIE Press, Bellingham, WA, pp. 95–148, 2004.
11. Bar-Cohen, Y., Kim, K. J., Choi, H. R. and Madden, J. D. W., "Electroactive polymer materials," *Smart Materials and Structures*, Vol. 16, Issue 2, 2007.
12. Bar-Cohen, Y. and Breazeal, C. (Eds.), *Biologically-inspired Intelligent Robots*, Vol. PM122, SPIE Press, Bellingham, WA, pp. 1–393, 2003.

13. Pelrine, R., Kornbluh, R., Joseph, J., Heydt, R., Pei, Q. and Chiba, S., “High-field deformation of elastomeric dielectrics for actuators,” *Materials Science and Engineering C*, Vol. 11, pp. 89–100, 2000.
14. Pelrine, R., Eckerle, J. and Chiba, S., “Review of artificial muscle approaches,” invited paper, *Proc. Third International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1992.
15. Pelrine, R., Kornbluh, R., Joseph, J. and Chiba, S., “Electrostriction of polymer films for microactuators,” *Proc. IEEE Tenth Annual International Workshop on Micro Electro Mechanical Systems*, Nagoya, Japan, pp. 238–243, 1997.
16. Zhang, Q., Bharti, V. and Zhao, X., “Giant electrostriction and relaxor ferroelectric behavior in electron-irradiated poly (vinylidene fluoride-trifluoroethylene) copolymer,” *Science*, Vol. 280, pp. 2101–2104, 1998.
17. Zheny, M., Scheinbeim, J. I., Lee, J. W. and Newman, B. A., “High field electrostrictive response of polymers,” *Journal of Polymer Sciences, Part B—Polymer Physics*, Vol. 32, pp. 2721–2731, 1994.
18. Shkel, Y. and Klingenberg, D., “Material parameters for electrostriction,” *Journal of Applied Physics*, Vol. 80, Issue 8, pp. 4566–4572, 1996.
19. Furukawa, T. and Seo, N., “Electrostriction as the origin of piezoelectriciitity in ferroelectric polymers,” *Japanese Journal of Applied Physics*, Vol. 29, Issue 4, pp. 675–680, 1990.
20. Tobushi, H., Hayashi, S. and Kojima, S., “Mechanical properties of shape memory polymer of polyurethane series,” *JSME International Journal*, Series I, Vol. 35, Issue 3, 1992.
21. Baughman, R., Shacklette, L., Elsenbaumer, R., Pichta, E. and Becht, C., “Conducting polymer electromechanical actuators,” in J. L. Bredas and R. R. Chance (Eds.), *Conjugated Polymeric Materials: Opportunities in Electronics, Optoelectronics and Molecular Electronics*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 559–582, 1990.
22. De Rossi, D. and Chiarelli, P., “Biomimetic macromolecular actuators, macro-ion characterization, Chapter 40,” *American Chemical Society Symposium Series*, Vol. 548, pp. 517–530, 1994.
23. Oguro, K., Kawami, Y. and Takenaka, H., “Bending of an ion-conducting polymer film-electrode composite by an electric stimulus at low voltage,” *Journal of the Micromachine Society*, Vol. 5, pp. 27–30, 1992.
24. Shahinpoor, M., “Micro-electro-mechanics of ionic polymer gels as electrically controllable artificial muscles,” *Journal of Intelligent Material Systems and Structures*, Vol. 6, pp. 307–314, 1995.
25. Smela, E., Inganas, O. and Pei, Q., “Electrochemical muscles: Micromachining fingers and corkscrews,” *Advanced Materials, Communications Section*, Vol. 5, Issue 9, pp. 630–632, 1993.
26. Pei, Q., Inganas, O. and Lundstrom, I., “Bending layer strips built from polyaniline for artificial electrochemical muscles,” *Smart Materials and Structures*, Vol. 2, pp. 1–6, 1993.
27. Bobbio, S., Kellam, M., Dudley, B., Goodwin Johansson, S., Jones, S., Jacobson, J., Tranjan, F. and DuBois, T., “Integrated force arrays,” *Proc. IEEE Micro Electro Mechanical Systems Workshop*, Fort Lauderdale, FL, 1993.
28. Ashley, S., “Artificial muscles,” *Scientific American*, Vol. 289, Issue 4, pp. 52–59, 2003.
29. Niklaus, M., Rosset, S., Dubois, P. and Shea, H., “Ion-implanted compliant electrodes used in dielectric electroactive polymer actuators with large displacement,” *Procedia Chemistry*, Vol. 1, pp. 702–705, 2009.
30. Shahinpoor, M., Bar-Cohen, Y., Simpson, J. O. and Smith, J., “Ionic polymer-metal composites (IPMC) as biomimetic sensors and structures—A review,” *Smart Materials and Structures*, Vol. 7, pp. 15–30, 1998.
31. Shahinpoor, M., “Ionic polymer-conductor composites as biomimetic sensors: Robotic actuators and artificial muscles—A review,” *Electrochimica Acta*, Vol. 48, pp. 2343–2353, 2003.
32. Oguro, K., Takenaka, H. and Kawami, Y., “Actuator element,” U.S. Patent #5268082, 1993.
33. Shahinpoor, M., Adolf, D., Segalman, D. and Witkowski, W., “Electrically controlled polymeric gel actuators, U.S. Patent #5250167, 1993.
34. Asada, A., Oguro, K., Nishimura, Y., Misuhata, M. and Takenaka, H., “Bending of polyelectrolyte membrane-platinum composites by electric stimuli, I. Response characteristics to various wave forms,” *Polymer Journal*, Vol. 27, pp. 436–440, 1995.
35. Shahinpoor, M. and Mojarrad, M., “Soft actuators and artificial muscles,” U.S. Patent #6109852, 2000.

36. De Gennes, P. G., Okumura, K., Shahinpoor, M. and Kim, K. J., "Mechanoelectric effects in ionic gels," *Europhysics Letters*, Vol. 50, Issue 4, pp. 513–518, 2000.
37. De Silva, C. W., *Mechanics of Materials*, CRC Press/Taylor & Francis, Boca Raton, FL, 2014.
38. Pelrine, R., Kornbluh, R. and Joseph, J., "Electrostriction of polymer dielectrics with compliant electrodes as a means of actuation," *Sensors and Actuators A: Physical*, Vol. 64, pp. 77–85, 1998.
39. Daghlian, C. and Howard, L., *Scanning Electron Microscope Image of Lung Trachea Epithelium*, Dartmouth E. M. Facility, Dartmouth College, Hanover, NH, 2001. Available at http://remf.dartmouth.edu/images/mammalianLungSEM/source/9.html.
40. Zhou, Z. and Liu, Z., "Biomimetic cilia based on MEMS technology," *Journal of Bionic Engineering*, Vol. 5, pp. 358–365, 2008.
41. Suh, J. W., Glander, S. F., Darling, R. B., Storment, C. W. and Kovacs, G. T. A., "Organic thermal and electrostatic ciliary microactuator array for object manipulation," *Sensors and Actuators A*, Vol. 58, pp. 51–60, 1997.
42. Suh, J. W., Darling, R. B., Böhringer, K. F., Donald, B. R., Baltes, H. and Kovacs, G. T. A., "CMOS integrated organic ciliary array as a general-purpose micromanipulation tool for small objects," *Journal of Microelectromechanical Systems*, Vol. 8, pp. 483–496, 1999.
43. Terry, M., Reiter, J. M., Böhringer, K. F., Suh, J. W. and Kovacs, G. T. "Docking system for microsatellites based on MEMS actuator arrays," *IOP Journal of Smart Materials and Structures*, Vol. 10, pp. 1176–1184, 2001.
44. Chen, Y. M., Suh, J. W., Kovacs, G. T., Darling, R. B. and Böhringer, K. F., "Modeling and control of a 3-Degree-of-Freedom walking microrobot," *Hilton Head 2006: A Solid State Sensor, Actuator, and Microsystems Workshop*, Hilton Head Island, NC, 2006.
45. den Toonder, J. M. J., Bos, F. M., Broer, D. J., Filippini, L., Gillies, M., de Goede, J., Mol, T. et al., "Artificial cilia for active microfluidic mixing," *Lab on a Chip*, Vol. 8, pp. 533–541, 2008.
46. den Toonder, J. M. J. and Onck, P. R., "Microfluidic manipulation with artificial/bioinspired cilia," *Trends in Biotechnology*, Vol. 31, Issue 2, pp. 85–91, 2013.
47. Evans, B. A., Shields, A. R., Carroll, R. L., Washburn, S., Falvo, M. R. and Superfine, R., "Magnetically actuated nanorod arrays as biomimetic cilia," *Nano Letters*, Vol. 7, pp. 1428–1434, 2007.
48. P2SI 900HT Tech Sheet, Performance Polymer Solutions Inc. 2711 Lance Drive, Moraine, OH 45409.
49. Smith, S. B., Finzi, L. and Bustamante, C., "Direct mechanical measurements of the elasticity of single DNA molecules by using magnetic beads," *Science*, Vol. 258, pp. 1122–1126, 1992.
50. Lee, G. U., Chrisey, L. A. and Colton, R. J., "Direct measurement of the forces between complementary strands of DNA," *Science*, Vol. 266, pp. 771–773, 1994.
51. Cluzel, P., Lebrun, A., Heller, C., Lavery, R., Viovy, J.-L., Chatenay, D. and Caron, F., "DNA: An extensible molecule," *Science*, Vol. 271, pp. 792–794, 1996.
52. Smith, S. B., Cui, Y. and Bustamante, C., "Overstretching B-DNA: The elastic response of individual double-stranded and single-stranded DNA molecules," *Science*, Vol. 271, pp. 795–799, 1996.
53. Collard, D., Yamahata, C., Legrand, B., Takekawa, T., Kumemura, M., Sakaki, N., Hashiguchi, G. and Fujita, H., "Towards mechanical characterization of biomolecules by MNEMS tools," *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 2, Issue 3, pp. 262–271, 2007.
54. Kada, G., Kienberger, F. and Hinterdorfer, P., "Atomic force microscopy in bionanotechnology," *Nanotoday*, Vol. 3, Issue 1–2, pp. 13–19, 2008.
55. Bustamante, C., Bryant, Z. and Smith, S. B., "Ten years of tension: Single molecule DNA mechanics," *Nature*, Vol. 421, pp. 423–427, 2003.
56. Strick, T. R., Allemand, J. F., Bensimon, D., Bensimon, A. and Croquette, V., "The elasticity of a single supercoiled DNA molecule," *Science*, Vol. 271, pp. 1835–1837, 1996.
57. Bancaud, A., Conde e Silva, N., Barbi, M., Wagner, G., Allemand, J.-F., Mozziconacci, J., Lavelle, C. et al., "Structural plasticity of single chromatin fibers revealed by torsional manipulation," *Nature Structural & Molecular Biology*, Vol. 13, pp. 444–450, 2006.
58. Strick, T. R., Croquette, V. and Bensimon, D., "Single-molecule analysis of DNA uncoiling by a type II topoisomerase," *Nature*, Vol. 404, pp. 901–904, 2000.

59. Gore, J., Bryant, Z., Stone, M. D., Nollmann, M., Cozzarelli, N. R. and Bustamante, C., "Mechanochemical analysis of DNA gyrase using rotor bead tracking," *Nature*, Vol. 439, pp. 100–104, 2006.
60. Clausen-Schaumann, H., Rief, M., Tolksdorf, C. and Gaub, H. E., "Mechanical stability of single DNA molecules," *Biophysical Journal*, Vol. 78, pp. 1997–2007, 2000.
61. Hards, A., Zhou, C. Q., Seitz, M., Brauchle, C. and Zumbusch, A., "Simultaneous AFM manipulation and fluorescence imaging of single DNA strands," *Chemphyschem*, Vol. 6, pp. 534–540, 2005.
62. Neuman, K. C. and Block, S. M., "Optical trapping." *Review of Scientific Instruments*, Vol. 75, pp. 2787–2809, 2004.
63. Ozkan, M., Wang, M., Ozkan, C., Flynn, R. and Esener, S., "Optical manipulation of objects and biological cells in microfluidic devices," *Biomedical Microdevices*, Vol. 5, Issue 1, pp. 61–67, 2003.
64. Wuite, G. J. L., Smith, S. B., Young, M., Keller, D. and Bustamante, C., "Single-molecule studies of the effect of template tension on T7 DNA polymerase activity," *Nature*, Vol. 404, pp. 103–106, 2000.
65. Brower-Toland, B. D., Smith, C. L., Yeh, R. C., Lis, J. T., Peterson, C. L. and Wang, M. D., "From the cover: Mechanical disruption of individual nucleosomes reveals a reversible multistage release of DNA," *Proceedings of the National Academy of Sciences*, Vol. 99, pp. 1960–1965, 2002.
66. Gosse, C. and Croquette, V., "Magnetic tweezers: Micromanipulation and force measurement at the molecular level," *Biophysical Journal*, Vol. 82, pp. 3314–3329, 2002.
67. Hashiguchi, G., Goda, T., Hosogi, M., Hirano, K., Kaji, N., Baba, Y., Kakushima, K. and Fujita, H., "DNA manipulation and retrieval from an aqueous solution with micromachined nanotweezers," *Analytical Chemistry*, Vol. 75, pp. 4347–4350, 2003.
68. Yamahata, C., Collard, D., Legrand, B., Takekawa, T., Kumemura, M., Hashiguchi, G. and Fujita, H., "Silicon nanotweezers with subnanometer resolution for the micromanipulation of biomolecules," *Journal of Microelectromechanical Systems*, Vol. 17, Issue 3, pp. 623–631, 2008.
69. Yamahata, C., Collard, D., Takekawa, T., Kumemura, M., Hashiguchi, G. and Fujita, H., "Humidity dependence of charge transport through DNA revealed by silicon-based nanotweezers manipulation," *Biophysical Journal*, Vol. 94, Issue 1, pp. 63–70, 2008.
70. Yamahata, C., Sarajlic, E., Jalabert, L., Kumemura, M., Collard, D. and Fujita, H., "Mechanical characterization of biomolecules in liquid using silicon tweezers with subnanonewton resolution," *IEEE 22nd International Conference on Micro Electro Mechanical Systems, MEMS 2009*, pp. 607–610, 2009.
71. Lafitte, N., Haddab, Y., Le Torrec, Y., Guillou, H., Kumemura, M., Jalabert, L., Collard, D. and Fujita, H., "Improvement of silicon nanotweezers sensitivity for mechanical characterization of biomolecules using closed- loop control," *IEEE/ASME Transactions on Mechatronics*, Vol. 20, Issue 3, pp. 1418–1427, 2015.
72. Kumemura, M., Tamura, K., Hashiguchi, G. and Fujita, H., "DNA molecular isolation in nano channel for single molecule trapping between micro electrodes," *Proceedings of 9th International Conference on Miniaturized Systems for Chemistry and Life Sciences, Microtas 2005*, Vol. 9, pp. 521–523, 2005.
73. Sieber, A., Valdastri, P., Houston, K., Eder, C., Tonet, O., Menciassi, A. and Dario, P., "A novel haptic platform for real time bilateral biomanipulation with a MEMS sensor for triaxial force feedback," *Sensors and Actuators A*, Vol. 142, pp. 19–27, 2008.
74. Kim, D.-H., Pak, K. W., Park, J., Levchenko, A. and Sun, Y., "Microengineered platforms forcell mechanobiology," *Annual Review of Biomedical Engineering*, Vol. 11, Issue 1, pp. 203–233, 2009.
75. Altman, G. H., Horan, R. L., Martin, I., Farhadi, J., Stark, P. R., Volloch, V., Richmond, J. C., Vunjak-Novakovic, G. and Kaplan, D. L., "Cell differentiation by mechanical stress," *FASEB Journal*, Vol. 16, Issue 2, pp. 270–272, 2002.
76. Lee, D. A., Knight, M. M., Campbell, J. J. and Bader, D. L., "Stem cell mechanobiology," *Journal of Cellular Biochemistry*, Vol. 112, Issue 1, pp. 1–9, 2011.
77. Yang, S. and Saif, T., "Micromachined force sensors for the study of cell mechanics," *Review of Scientific Instruments*, Vol. 76, Issue 4, 2005.

78. Serrell, D. B., Oreskovic, T. L., Slifka, A. J., Mahajan, R. L. and Finch, D. S., "A uniaxial bioMEMS device for quantitative force–displacement measurements," *Biomedical Microdevices*, Vol. 9, Issue 2, pp. 267–275, 2007.
79. Erismis, M. A., Pereira Neves, H., De Moor, P., Puers, R. and Van Hoof, C., "A water-tight packaging of MEMS electrostatic actuators for biomedical applications," *Microsystem Technologies*, Vol. 16, Issue 12, pp. 2109–2113, 2010.
80. Hausherr, T. C., Majd, H., Joss, D., Müller, A., Pioletti, D. P., Gijs, M. A. M. and Yamahata, C., "Capillary-valve-based platform towards cell-on-chip mechanotransduction assays," *Sensors and Actuators B*, Vol. 188, pp. 1019–1025, 2013.
81. Bruus, H., *Theoretical Microfluidics, Oxford Master Series in Physics*, Chapter 7, Oxford University Press, Oxford, New York, 2008.
82. Man, P. F., Mastrangelo, C. H., Burns, M. A. and Burke, D. T., "Microfabricated capillarity-driven stop valve and sample injector," *Proc. 11th IEEE Annu. Int. Workshop Micro Electro Mech. Syst.*, pp. 45–50, 1998.
83. Chen, J. M., "Analysis and experiment of capillary valves for microfluidics on a rotating disk," *Microfluidics and Nanofluidics*, Vol. 4, Issue 5, pp. 427–437, 2008.
84. Tsai, N.-C. and Sue, C.-Y., "Review of MEMS-based drug delivery and dosing systems," *Sensors and Actuators A*, Vol. 134, pp. 555–564, 2007.
85. Tao, S. L. and Desai, T. A., "Microfabricated drug delivery systems: From particles to pores," *Advanced Drug Delivery Reviews*, Vol. 55, pp. 315–328, 2003.
86. Shawgo, R. S., Richards Grayson, A. C., Li, Y. and Cima, M. J., "BioMEMS for drug delivery," *Current Opinion in Solid State and Materials, Science*, Vol. 6, pp. 329–334, 2002.
87. Nisar, A., Afzulpurkar, N., Mahaisavariya, B. and Tuantranont, A., "MEMS-based micropumps in drug delivery and biomedical applications," *Sensors and Actuators B*, Vol. 130, pp. 917–942, 2008.
88. Zengerle, R., Ulrich, J., Kluge, S., Richter, M. and Richter, A., "A bidirectional silicon micropump," *Sensors and Actuators A: Physical*, Vol. 50, pp. 81–86, 1995.
89. Zengerle, R., Richter, M. and Sandmaier, H., "A micro membrane pump with electrostatic actuation," *Proceedings of IEEE, Micro Electro Mechanical System*, pp. 19–24, 1992.
90. Cabuz, C., Herb, W. R., Cabuz, E. I. and Lu, S. T., "The dual diaphragm pump," *Proceedings of the IEEE MEMS 2001*, pp. 519–522, 2001.
91. Judy, J. W., Tamagawa, T. and Polla, D. L., "Surface micromachined micropump," *Proc. MEMS 91*, pp. 182–186, 1991.
92. Koch, M., Harris, N., Evans, A. G. R., White, N. M. and Brunnschweiler, A., "A novel micromachined pump based on thick-film piezoelectric actuation," *Sensors and Actuators A: Physical*, Vol. 70, pp. 98–103, 1998.
93. Didier, M., Harald, V. L., Gilles, R. M. and Roland, H., "High-performance silicon micropump for an implantable drug delivery system," *Proceedings of the IEEE MEMS 99*, pp. 541–546, 1999.
94. Schabmueller, C. G. J., Koch, M., Mokhtari, M. E., Evans, A. G. R., Brunnschweiler, A. and Sehr, H., "Self-aligning gas/liquid micropump," *Journal of Micromechanics and Microengineering*, Vol. 12, pp. 420–424, 2000.
95. Bohm, S., Olthuis, W. and Bergveld, P., "A plastic micropump constructed with conventional techniques and materials," *Sensors and Actuators A: Physical*, Vol. 77, pp. 223–228, 1999.
96. Gass, V., Vanderschoot, B. H., Jeanneret, S. and Derooij, N. F., "Integrated flow regulated silicon micropump," *Sensors and Actuators A*, Vol. 43, pp. 335–338, 1994.
97. Van Lintel, H. T. G., van De Pol, F. C. M. and Bouwstra, S., "A piezoelectric micropump based on micromachining of silicon," *Sensors and Actuators*, Vol. 15, Issue 2, pp. 153–167, 1988.
98. Junwu, K., Zhigang, Y., Taijiang, P., Guangming, C. and Boda, W., "Design and test of a high performance piezoelectric micropump for drug delivery," *Sensors and Actuators A: Physical*, Vol. 121, pp. 156–161, 2005.
99. Lee, S. W., Sim, W. Y. and Yang, S. S., "Fabrication and *in vitro* test of a microsyringe," *Sensors and Actuators A: Physical*, Vol. 83, pp. 17–23, 2000.
100. Feng, G. H. and Kim, E. S., "Piezoelectrically actuated dome-shaped diaphragm micropump," *Journal of Microelectromechanical Systems*, Vol. 14, pp. 192–199, 2005.

101. Geipel, A., Doll, A., Goldschmidtboing, F., Jantscheff, P., Esser, N., Massing, U. and Woias, P., "Pressure independent micropump with piezoelectric valves for low flow drug delivery systems," *MEMS2006*, Istanbul, Turkey, January 2006.
102. Ma, B., Sheng, L., Gan, Z., Liu, G., Cai, X., Zhang, H. and Yang, Z., "A PZT insulin pump integrated with a silicon micro needle array for transdermal drug delivery," *Proceedings of the 56th Electronic Components and Technology Conference*, pp. 677–681, 2006.
103. Hsu, Y. C., Lin, S. J. and Hou, C. C., "Development of peristaltic antithrombogenic micropumps for *in vitro* and ex vivo blood transportation tests," *Microsystem Technologies*, Vol. 14, pp. 31–41, 2008.
104. Suzuki, T., Teramura, Y., Hata, H., Inokuma, K., Kanno, I., Iwata, H. and Kotera, H. "Development of a micro biochip integrated travelling wave micropumps and surface plasmon resonance imaging sensors," *Microsystem Technologies*, Vol. 13, pp. 1391–1396, 2007.
105. Linnemann, R., Woias, P., Senfft, C.-D. and Ditterich, J. A., "A self priming and bubble tolerant silicon micropump for liquids and gases," *Proceedings of the MEMS 98*, Heidelberg, Germany, pp. 532–537, 1998.
106. Morris, C. J. and Forster, F. K., "Optimization of a circular piezoelectric bimorph for a micropump driver," *Journal of Micromechanics and Microengineering*, Vol. 10, pp. 459–465, 2000.
107. Jeong, O. C. and Yang, S. S., "Fabrication and test of a thermopneumatic micropump with a corrugated p+ diaphragm," *Sensors and Actuators A: Physical*, Vol. 83, pp. 249–255, 2000.
108. Zimmermann, S., Frank, J. A., Liepmann, D. and Pisano, A. P., "A planar micropump utilizing thermopneumatic actuation and in-plane flap valves," *17th IEEE International Conference on Micro Electro Mechanical Systems (MEMS): Maastricht MEMS 2004 Technical Digest*, pp. 462–465, 2004.
109. Schomburg, W. K., Vollmer, J., Bustgens, B., Fahrenberg, J., Hein, H. and Menz, W., "Microfluidic components in LIGA technique," *Journal of Micromechanics and Microengineering*, Vol. 4, pp. 186–191, 1994.
110. Van De Pol, F. C. M., Van Lintel, H. T. G., Elwenspoek, M. and Fluitman, J. H. J., "A thermopneumatic micropump based on microengineering techniques," *Sensors and Actuators A: Physical*, Vol. 21, pp. 198–202, 1990.
111. Zdelblick, M. J. and Angell, J. B., "A microminiature electric to fluidic valve," *Proceedings of the 4th International Conference Solid State Sensors and Actuators (Transducer '87)*, Tokyo, Japan, p. 827, 1987.
112. Hwang, S. R., Sim, W. Y., Jeon, D. H., Kim, G. Y., Yang, S. S. and Pak, J. J., "Fabrication and test of a submicroliter level thermopneumatic micropump for transdermal drug delivery," *Proceedings of the 3rd Annual International IEEE EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, Kahuku, Oahu, Hawaii, 2005.
113. Kim, J. H., Na, K. H., Kang, C. J. and Kima, Y. S., "A disposable thermopneumatic actuated micropump stacked with PDMS layers and ITO coated glass," *Sensors and Actuators A: Physical*, Vol. 120, pp. 365–369, 2005.
114. Jeong, O. C., Park, S. W., Yang, S. S. and Pak, J. J., "Fabrication of a peristaltic PDMS micropump," *Sensors and Actuators A: Physical*, Vol. 123/124, pp. 453–458, 2005.
115. Xu, D., Wang, L., Ding, G., Zhou, Y., Yu, A. and Cai, B., "Characteristics and fabrication of NiTi/Si diaphragm micropump," *Sensors and Actuators A: Physical*, Vol. 93, pp. 87–92, 2001.
116. Benard, W. L., Kahn, H., Heuer, A. H. and Huff, M. A., "A titanium–nickel shape-memory alloy actuated micropump," *Proceedings of the International Conference on Solid-State Sensors and Actuators*, Vol. 1, pp. 361–364, 1997.
117. Benard, W. L., Kahn, H., Heuer, A. H. and Huff, M. A., "Thin-film shape-memory alloy actuated micropumps," *Journal of Microelectromechanical Systems*, Vol. 7, pp. 245–251, 1998.
118. Shuxiang, G. and Fukuda, T., "SMA actuator based novel type of micropump for biomedical application," *2004 IEEE International Conference*, Vol. 2, pp. 1616–1621, 2004.
119. Zhan, C., Lo, T., Liu, L. and Tsien, P., "A silicon membrane micropump with integrated bimetallic actuator," *Chinese Journal of Electronics*, Vol. 5, pp. 29–35, 1996.

120. Yang, Y., Zhou, Z., Ye, X. and Jiang, X., "Bimetallic thermally actuated micropump," *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC*, Vol. 59, pp. 351–354, 1996.
121. Zou, J. X., Ye, Y. Z., Zhou, Y. and Yang, Y., "A novel thermally actuated silicon micropump," *Proceedings of the 1997 International Symposium on Micromechatronics and Human Science*, pp. 231–234, 1997.
122. Guo, S., Nakamura, T., Fukuda, T. and Oguro, K., "Design and experiments of micropump using ICPF actuator," *Proceedings of the Seventh International IEEE Symposium on Micro Machine and Human Science*, pp. 235–240, 1996.
123. Guo, S. and Asaka, K., "Polymer based new type of micropump for biomedical applications," *Proceedings of the IEEE Conference on Robotics & Automation*, Taipei, Taiwan, pp. 1830–1835, 2003.
124. Guo, S. and Fukuda, T., "Development of the micro pump using ICPF actuator," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 266–271, 1997.
125. Guo, S., Sugimoto, K., Fukuda, T. and Oguro, K., "New type of capsule medical micropump," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 55–60, 1999.
126. Gong, Q., Zhou, Z., Yang, Y. and Wang, X., "Design, optimization and simulation on microelectromagnetic pump," *Sensors and Actuators A: Physical*, Vol. 83, pp. 200–207, 2000.
127. Yamahata, C., Lotto, C., Al Assaf, E. and Gijs, M. A. M. "A PMMA valveless micropump using electromagnetic actuation," *Microfluid Nanofluid*, Vol. 1, pp. 197–207, 2005.
128. Yamahata, C., Chastellain, M., Parashar, V. K., Petri, A., Hofmann, H. and Gijs, M. A. M., "Plastic Micropump with ferrofluidic actuation," *Journal of Microelectromechanical Systems*, Vol. 14, pp. 96–102, 2005.
129. Pan, T., McDonald, S. J., Kail, E. M. and Ziaiel, B., "A magnetically driven PDMS micropump with ball check valves," *Journal of Micromechanics and Microengineering*, Vol. 15, pp. 1021–1026, 2005.
130. Kim, J., Khetarpal, I., Sen, S. and Murray, R. M., "Synthetic circuit for exact adaptation and fold-change detection," *Nucleic Acids Research*, Vol. 42, Issue 9, pp. 6078–6089, 2014.
131. Siegal-Gaskins, D., Tuza, Z. A., Kim, J., Noireaux, V. and Murray, R. M., "Resource usage and gene circuit performance characterization in a cell-free 'breadboard'," *ACS Synthetic Biology*, Vol. 3, Issue 6, pp. 416–425, 2014.
132. Sun, Z. Z., Yeung, E., Hayes, C. A., Noireaux, V. and Murray, R. M., "Linear DNA for rapid prototyping of synthetic biological circuits in an Escherichia coli based TX-TL cell-free system," *ACS Synthetic Biology*, Vol. 3, Issue 6, pp. 387–397, 2014.
133. Siegal-Gaskins, D., Noireaux, V. and Murray, R. M., "Biomolecular resource utilization in elementary cell-free gene circuits," *American Control Conference (ACC)*, Washington, DC, June 17–19, 2013.
134. Tuza, Z. A., Singhal, V., Kim, J. and Murray, R. M., "An in silico modeling toolbox for rapid prototyping of circuits in a biomolecular "breadboard" system," *52nd IEEE Conference on Decision and Control*, Florence, Italy, December 10–13, 2013.
135. Fernandez, R. E., Stolyarova, S., Chadha, A., Bhattacharya, E. and Nemirovsky, Y., "MEMS composite porous silicon/polysilicon cantilever sensor for enhanced triglycerides biosensing," *IEEE Sensors Journal*, Vol. 9, Issue 12, pp. 1660–1666, 2009.
136. Chen, C., Hwang, R., Huang, L., Lin, S., Chen, H., Yang, Y., Lin, Y. et al., "A wireless bio-MEMS sensor for C-reactive protein detection based on nanomechanics," *IEEE Transaction on Biomedical Engineering*, Vol. 56, Issue 2, pp. 462–470, 2009.
137. Huang, X., Li, S., Schultz, J., Wang, Q. and Lin, Q., "A capacitive MEMS viscometric sensor for affinity detection of glucose," *Journal of Microelectromechanical Systems*, Vol. 18, Issue 6, pp. 1246–1254, 2009.
138. Kudo, H., Sawada, T., Kazawa, E., Yoshida, H., Iwasaki, Y. and Mitsubayashi, K., "A flexible and wearable glucose sensor based on functional polymers with Soft-MEMS techniques," *Biosensors and Bioelectronics*, Vol. 22, pp. 558–562, 2006.
139. Huang, X., Li, S., Schultz, J. S., Wang, Q. and Lina, Q., "A MEMS affinity glucose sensor using a biocompatible glucose-responsive polymer," *Sensors and Actuators B*, Vol. 140, pp. 603–609, 2009.

140. Kim, K., Cheng, J., Liu, Q., Wu, X. Y. and Sun, Y., “Investigation of mechanical properties of soft hydrogel microcapsules in relation to protein delivery using a MEMS force sensor,” *Journal of Biomedical Materials Research Part A*, Vol. 92, Issue 1, pp. 103–113, 2010.
141. Sokhanvar, S., Packirisamy, M. and Dargahi, J., “MEMS endoscopic tactile sensor: Toward in-situ and in-vivo tissue, softness characterization,” *IEEE Sensors Journal*, Vol. 9, Issue 12, pp. 1679–1687, 2009.
142. Satake, D., Ebi, H., Oku, N., Matsuda, K., Takao, H., Ashiki, M. and Ishida, M., “A sensor for blood cell counter using MEMS technology,” *Sensors and Actuators B*, Vol. 83, pp. 77–81, 2002.
143. Ko, S. C., Jun, C., Jang, W. I. and Choi, C., “Micromachined air-gap structure MEMS acoustic sensor using reproducible high-speed lateral etching and CMP process,” *Journal of Micromechanics and Microengineering*, Vol. 16, pp. 2071–2076, 2006.

7

System Identification in Human Adaptive Mechatronics

Bin Xi and Clarence W. de Silva

CONTENTS

7.1 From Manual Control to Human Adaptive Mechatronics 254
7.2 Human in the Loop 255
7.3 Classical HO Model 256
  7.3.1 Quasi-Linear Structure 257
  7.3.2 Crossover Model 257
7.4 Identification of Quasi-Linear Model 259
  7.4.1 Signal and Spectra 259
  7.4.2 Nonparametric Quasi-Linear Model 262
  7.4.3 Parametric Quasi-Linear Model 264
  7.4.4 Experiment and Model Identification Results 265
7.5 Identification through Optimal Control Theory 268
  7.5.1 Linear Regulator Problem 269
  7.5.2 LQG Controller without Time Delay 269
  7.5.3 LQG Controller with Time Delay 271
  7.5.4 Optimal Control Model for the Human Operator 274
  7.5.5 Human Optimal Control Model (OCM) 276
  7.5.6 Motor Noise Effect 278
  7.5.7 Modified Optimal Control Model (MOCM) 279
  7.5.8 Identification of Optimal Control Model 283
  7.5.9 Data-Based HO Model Identification 289
7.6 Conclusions 293
References 293

SUMMARY Human adaptive mechatronics is related to human–machine system design and targeted to enhance the overall human–machine performance through the interactive communication and decision making between humans and machines. An important issue in this field is the identification of a human operator model when the operator is in the loop to conduct some control task. Manual control systems have laid a solid foundation for the process of human operator modeling. In this chapter, a quasi-linear model and crossover model are reviewed first. A nonparametric frequency response model and parametric transfer function model are obtained in an experiment of compensatory tracking. This can be regarded as a frequency domain approach. Modern control theory can be adopted to carry out modeling in the time domain. The optimal control model and its modification are discussed in detail. The identifiability problem is also treated. In a disturbance

rejection experiment, an LQG controller cascaded with a deteriorating filter is developed to facilitate the system identification.

7.1 From Manual Control to Human Adaptive Mechatronics

Machine control by a human operator still exists even though highly automatic and intelligent mechatronic products are numerous. From aircraft to ground vehicles, a human operator (HO) may maneuver a stick, wheel, or pedal through neuromotor systems to operate the plant safely and conveniently. In process control, skilled manual control can make the overall process perform at the expected efficiency. In telemedicine and telesurgery, a skillful operator is necessary for the successful operation of the process. Of course, in all these cases, errors and mistakes caused by the HO can also degrade the system performance. Hence, it is plausible to assume that an HO plays a dominant role in human–machine systems on account of the inherent capabilities and limitations of the HO. The competence of HOs lies primarily in their ability to scan a visual signal and make perceptions and inferences on that basis. The brain system of an HO is an effective signal processing unit. Through their neuromuscular system HOs can perform actions on a machine subjected to environmental influences and internal psychological and physiological restrictions. All these functions make an HO analogous to an engineering controller with excellent learning and adaptive abilities with some weakness.

A HO does not exclusively dictate the performance of a human–machine system when an intelligent mechatronic system is incorporated. A mechatronic system may possess some intellectual potential and ability to react to an HO's action. If the mechatronic system makes proper adjustments in response to the HO's actions, the behavior of the overall human machine system will be enhanced. However, the state of the art is not adequate, and there is much room for improvement. A recent intensive study of human adaptive mechatronics (HAM) has strongly promoted this idea [1].

HAM is a relatively new concept, which was first proposed by Japanese scholars [2]. They sought to solve the problem of information asymmetry in the sense that an HO can extract information from a machine during its operation, but the machine cannot do the same concerning the HO. It is universally recognized that proper matching between the HO and machine is necessary for good performance. In this context, it is important for the machine to know the dynamics of the HO so that proper predictions and adjustments can be made on the machine. To this end, control engineering can play an important role. Specifically, the HO may be treated as a controller that interacts with the machine. This idea highlights the need for an HO model. Fortunately, since the 1940s much progress has been made in the research on manual control in applications of aeronautics and astronautics. Initially, pilot-vehicle dynamics was the primary concern. In this manner, manual control has laid a solid foundation for HAM.

However, there are many more disciplines involved in HAM research, ranging from conventional control engineering to artificial intelligence, psychology and physiology, information and decision theory, and so on. Among the many topics of HAM, this chapter only concentrates on the modeling issues, specifically, the system identification approaches in both time and frequency domains. The general system identification emphasizes finding a specific model to minimize the model error in some sense, which is primarily an optimization problem. But in the context of HAM, system identification is conducted in the process of carrying out some control task, and hence, control actions should be taken into account. With this understanding, the behavior of the human in the loop is examined now.

7.2 Human in the Loop

The simple human-in-the-loop setup is shown in Figure 7.1, in which we ignore the environmental effects on the loop. Four blocks constitute the entire loop. A plant is controlled by the HO through some control manipulators, a joystick for example, and the output of the plant is fed to a display system in which HOs can view all the necessary information to dictate their actions. In this manner, a closed loop is formed to meet the operational requirements.

It is quite plausible that a human-in-the-loop configuration is nothing more than the feedback control of a dynamic system except that the hard connections in the feedback channel in the general control loop is replaced by visual scanning. Hence, it is natural to tackle the HO problem by a cybernetic approach. Here, we emphasize that the HO as a controller is physiological rather than physical. These two types of controllers share many common characteristics but are distinguished from each other in many aspects, too.

A physical controller, composed of mechanical or electrical elements or computer software, can duplicate its actions under the same stimulus for a virtually unlimited number of times. It can be sufficiently robust, through careful design, to resist external influences. However, the HO as a physiological controller generally cannot repeat the same actions exactly even under the same stimulus. This is because due to fatigue, stress, and so on, an HO is vulnerable to outside influences. A physical controller can operate continuously for very long periods while an HO cannot function for more than one regular shift. Indeed, an HO can maintain attention only for a relatively short time.

These differences stem from the nature of an HO controller and a physical controller. A physical controller is typically fixed in structure and parameters in the absence of adaptive control. An HO, however, is not fixed and stationary. In the context of modeling, no matter how complex a physical controller is, its analytical (mathematical) model is available through its hardware or software implementation. An HO, however, is quite difficult to model in terms of the structure and parameters. In spite of these marked distinctions between an HO and a physical controller, mimicking of the traditional control engineering approach to analyze and model an HO can be desirable. Specifically, an HO has behavior that is analogous to a physical controller.

Why do we require an HO model? In the design stage of a mechatronic system in which an HO is involved, with an HO model at hand, we can consider how to specify the dynamic

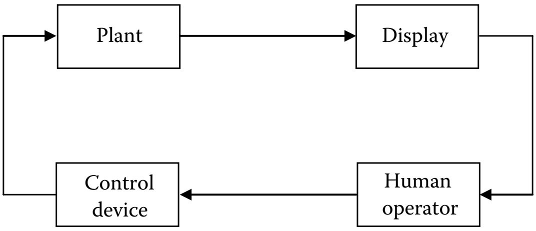


FIGURE 7.1
A human-in-the-loop arrangement.

performance of the controlled plant to allocate tasks among the HO and the controlled plant so that the overall closed-loop system performs properly. We can also rate the HO performance quantitatively by means of an HO model in order to develop an intelligent training system. With the models of both HO and the controlled plant, we can make an adaptive human–machine interface, which can lead to effective matching between the HO and the machine.

We may possess extensive experience in the analysis and synthesis of a physical controller in a closed-loop system. However, it is not trivial to adopt them in an HO controller. The behavior of an HO may be understood and expressed on the basis of control engineering, physiology, experimental psychology, information theory, and so on. To this end, modern HAM research is promising. Multidisciplinary knowledge is needed in HAM. Among the many research areas of HO, here we focus on the modeling issue. Subsequently, system identification methodology is discussed. This will form an important contribution in the development of an overall HAM system.

7.3 Classical HO Model

The layout of an HO control system is shown in Figure 7.2. The control task for the HO is to make the plant output y follow the reference input u as closely as possible. Through a display system, the HO can sense the tracking error and the error rate, which are the inputs to the HO. When only the error is observed, the control system is termed a single-loop compensatory system; when both reference input and error are observed, it is termed a pursuit tacking system [3].

The first impression this schematic diagram gives is that it is just a classical feedback control loop with the usual sensors and actuators located at the appropriate places. However, the essential difference lies in the fact that sensing and actuating are carried out by the HO rather than some electrical or mechanical device. The task allocation among all the blocks in the classical feedback loop is clearly configured and fixed during the operation of the control system. But in a human–machine system, task allocation between the HO and the machine is interactive and adaptive. The model of an HO depends on the reference input model, plant model, and the control task specifications. For a physical controller, only its input signal plays a major role. But there are many key variables in an HO control system. Four kinds of variables can be listed, as follows [4]:

1. Task variables: Reference input and disturbance input, controlled variables, displayed information on all the devices.
2. Environmental variables: Temperature, lighting, and air pressure.

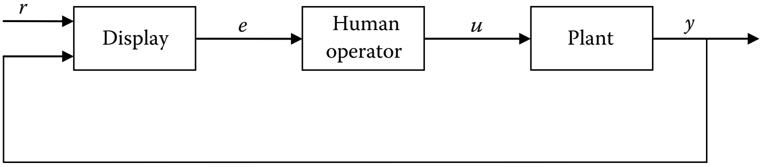


FIGURE 7.2
Closed-loop HO system.

3. Procedural variables: Variables associated with the experimental design and measurement procedure.
4. Operator-centered variables: Variables associated with training, motivation, and fatigue.

The working condition of an HO is so complex that highly nonlinear and nonstationary properties are unavoidable. But in the present chapter, a simple description for the HO model is provided. In particular, a linear time-invariant model is mathematically tractable and amenable in the engineering application. In what follows, we explore some classical HO models in the context of manual control systems.

7.3.1 Quasi-Linear Structure

A linear model is an approximation to a real nonlinear system. It should be noted that the behavior of an HO is quite complex with salient nonlinear and nonstationary properties. But as an analog of a conventional controller, some salient characteristics exist in the operation of an HO. While seeking to capture these characteristics, a linear model is a good choice from the viewpoints of analysis and identification. From the decomposition point of view, we can regard the output of an HO to consist of two parts. One part is linearly related to the input of the HO. The other part cannot be accounted for by the linear part and represents the lumped effects from nonlinearity and randomness. The quasi-linear model, pioneered by McRuer et al. [5,6], embodies this idea. It laid a solid foundation for the research in human–machine systems. It is examined and validated through experiments and observations. From the measured experimental data, it has been found that the response of an HO to the input signal with multifrequency components tends to follow the low-frequency components and considerably suppress the high-frequency components. Figure 7.3 shows the structure of the quasi-linear model. Here $e(t)$ is the input signal to the HO, usually the tracking error or error rate, $u'(t)$ is the output signal of the HO that is linearly related with $e(t)$, and $n(t)$ is a noise signal accounting for nonlinear and varying environmental effects, which is independent of $e(t)$; the term "remnant" is usually applied to denote this signal. The ultimate output of the HO is $u(t) = u'(t) + n(t)$.

The structure of the quasi-linear model opens the way to use classical and modern control theory to study an HO system. Only if the models for the HO and remnant statistics are available can we capture the behavior of the HO. This should be conducted in a closed loop, in which the controlled plant and the reference input together will determine the HO model.

7.3.2 Crossover Model

In a series of experiments for manual control, a highly trained operator carries out the control actions using a suitable manipulator. For compensatory tracking as the control task, different dynamic plants are studied to find the underlying HO model. McRuer and

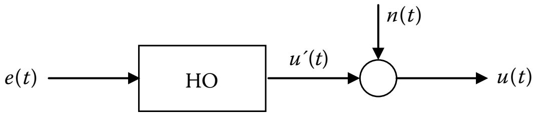


FIGURE 7.3
Quasi-linear model.

Krendel [4] studied some simple but essential dynamics and made the following observations. For the dynamic plant, K/s, an integrator by which we can simulate plane attitude control, an approximated HO model is $K_h e^{-\tau s}$, where K_h is an adjustable gain, and τ is an adjustable time delay; for the dynamic plant K/s^2, a second-order system by which we may simulate automobile longitudinal position control, an approximate HO model is $K_h s e^{-\tau s}$, and so on.

Based on these observations, McRuer et al. concluded that in the setting of a compensatory tracking task in which an HO is required to nullify the tracking error through manual operation, the transfer function for the HO can be expressed as [6]

$$H(s) = \left\{ K \frac{T_{\mathrm{L}} s + 1}{T_{\mathrm{I}} s + 1} \right\} \frac{e^{-s\tau}}{T_{\mathrm{N}} s + 1} \tag{7.1}$$

where K is loop gain, which is adjustable by the HO over some range in terms of display and units of control; τ is the time delay due to the reaction time, typically 0.12 to 0.20 sec; T_{L} is the adjustable lead time constant; T_{I} is the adjustable lag time constant; and T_{N} accounts for neuromotor dynamics. In Equation 7.1, the outside part of the bracket accounts for the HO's limitation on the total effect of reaction time. The inside part of the bracket represents the HO's equalization ability to fulfill the control task.

When an HO is in the loop, the loop transfer function is given by $H(s)P(s)$, where $P(s)$ is the transfer function of the controlled plant. To demonstrate the overall performance of the HO, a crucial parameter named crossover frequency ω_c is considered. At this frequency, the open loop transfer function satisfies

$$|H(i\omega_c)P(i\omega_c)| = 1 \tag{7.2}$$

Also, near this frequency, the open loop transfer function may be characterized by

$$H(i\omega)P(i\omega) = \frac{\omega_c}{i\omega} e^{-i\omega\tau} \tag{7.3}$$

This is the well-known crossover model, which is parameterized by the adjustable ω_c and τ. They are functions of the spectrum of the input signal and the dynamics of the controlled plant or are functions of task variables. It appears to be an integrator; hence, a good servo mechanism can be generated. In the low-frequency band $\omega < \omega_c$, the loop gain is high, and the desired output is almost the same as the input. In the high-frequency band $\omega > \omega_c$, the loop gain is low to guarantee robustness in performance, and feedback is almost inoperative. Therefore, the crossover frequency ω_c is the performance boundary of the closed-loop system. In the vicinity of the crossover frequency ω_c, we can establish the HO model as

$$H(i\omega) = \frac{\omega_c}{i\omega P(i\omega)} e^{-i\omega\tau} \tag{7.4}$$

This explicitly indicates that the HO is adaptive to the controlled plant and other task variables. For example, if the controlled plant has third-order dynamics, then $H(j\omega)$ should encompass two lead terms. Crossover frequency ω_c and the time delay τ are two adjustable parameters to reflect the HO's adaptability and optimality in a closed-loop system.

With this result in hand, we can approach the HO model by finding ω_c and the time delay τ that best fits the open loop transfer function with Equation 7.3, and then obtain the HO model by using Equation 7.4. However, this is just the local model in the vicinity of ω_c. For a wider frequency band model, we can take advantage of modern signal processing and system identification technologies. Furthermore, the HO model cannot be easily derived from experimental data if we stick to the crossover model structure for the loop transfer function model. Next, we will concentrate on the classical system identification methods.

7.4 Identification of Quasi-Linear Model

The quasi-linear model highlights the framework for the HO model, namely a linear component connected to input with a stochastic remnant component added to the output of the linear component. The crossover model specifies a predetermined loop transfer function model with a number of unknown parameters. The identification of the HO model uses a trial-and-error method to find the parameters of the crossover model. This might need many iterations. Other than identification of the direct crossover model, there are two other routes that can lead us to the HO model:

1. Direct identification of the HO model from the measurements of the error signal and the output signal of HO.
2. By putting the identification in the closed-loop form, the HO model may be obtained indirectly.

In all these methods, we can employ both nonparametric and parametric model paradigms. In the nonparametric form, a frequency response function is identified, and in the parametric form, a transfer function model with a finite number of parameters is identified. In order to perform the system identification task, we need some basic background knowledge on elementary liner, time-invariant systems and stochastic processes.

A fundamental characteristic of a linear time-invariant system is its impulse response function, which is essentially a time-domain model in nonparametric form. Its Fourier or Laplace transform is the transfer function, which is the frequency-domain model in nonparametric form. Complement to the nonparametric model, we also have the structured parametric model to describe a liner system, which needs parameter estimation to find the model.

In this section, we take advantage of statistical signal processing methods to carry out the identification of the HO model in nonparametric frequency domain form and parametric linear system form in the time domain. The crux for statistical signal processing is special types of signal transform techniques, of which Fourier transform is primary. Signal spectral estimation can be carried out by discrete-time Fourier transform (DFT), and the HO model can be obtained through signal spectral estimation. We begin with some elementary definitions in linear system theory and random processes that are related to HO modeling.

7.4.1 Signal and Spectra

It is well known that any time signal can be approximated by a linear combination of sinusoidal signals with different frequencies and amplitudes (also, phase angles, if the complex form is not used in the representation). The frequency content of a time domain signal can be revealed by its corresponding spectral description, which is characterized by the spectral density and classified as energy spectral density and power spectral density in terms of finite energy and finite power signals. Spectral theory can be applied to both deterministic and stochastic signals.

For a time signal $x(t)$, we use $X(\omega)$ to denote its Fourier transform. If $x(t)$ is a stochastic process that satisfies stationary condition [7], than its mean value is constant:

$$E\{x(t)\} = \mu_x \tag{7.5}$$

its auto-covariance function is given by

$$R_{xx}(\tau) = E\{(x(t) - \mu_x)(x(t - \tau) - \mu_x)\} \tag{7.6}$$

where τ is the time lag. Loosely speaking, auto-covariance measures the degree of dependence between signals with time lag τ. Generally, the larger the time lag τ, the weaker the dependence. The formal definition of the spectral density of $x(t)$ is

$$\Phi_x(\omega) = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-i\omega\tau}\,\mathrm{d}\tau \tag{7.7}$$

This is the Fourier transform of the auto-covariance function. For another stationary signal $y(t)$, the cross-covariance function between $x(t)$ and $y(t)$ can be defined as

$$R_{xy}(\tau) = E\{(x(t) - \mu_x)(y(t - \tau) - \mu_y)\} \tag{7.8}$$

Then the cross-spectral density between $x(t)$ and $y(t)$ is defined by

$$\Phi_{xy}(\omega) = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-i\omega\tau}\,\mathrm{d}\tau \tag{7.9}$$

This is the Fourier transform of the cross-covariance function. It should be noted that $\Phi_{xy}(\omega) = 0$ if $x(t)$ and $y(t)$ are uncorrelated with each other. Under these definitions, we now explore the relationship between signal spectra and a linear time-invariant system. For a linear system with input $u(t)$ and output $y(t)$, the corresponding frequency domain description is

$$Y(\omega) = G(i\omega)U(\omega) + W(\omega) \tag{7.10}$$

where $G(i\omega)$ is the system function or transfer function, which is the Fourier transform of its impulse response function; $W(\omega)$ is the Fourier transform of the disturbance signal $w(t)$, which accounts for the unexplained part by $u(t)$ through $G(i\omega)$, which is uncorrelated with the control input $u(t)$. The following two important relations hold [7,8]:

$$\Phi_y(\omega) = |G(i\omega)|^2\, \Phi_u(\omega) + \Phi_w(\omega) \tag{7.11}$$

and

$$\Phi_{yu}(\omega) = G(i\omega)\, \Phi_u(\omega) \tag{7.12}$$

From these results, we can identify the transfer function by

$$G(i\omega) = \frac{\Phi_{yu}(\omega)}{\Phi_u(\omega)} \tag{7.13}$$

The noise spectral density can be identified from

$$\Phi_w(\omega) = \Phi_y(\omega) - |G(i\omega)|^2\, \Phi_u(\omega) \tag{7.14}$$

This approach is called nonparametric identification because there is no explicit model structure specification and a finite number of model parameters. When we use sampled data, the continuous model $G(i\omega)$ is changed to $G_T(e^{i\omega T})$ [7], where T stands for the uniform sample interval. Then the same relations carry over as

$$\Phi_y(\omega) = |G_T(e^{i\omega T})|^2\, \Phi_u(\omega) + \Phi_w(\omega) \tag{7.15}$$

and

$$\Phi_{yu}(\omega) = G(e^{i\omega T})\, \Phi_u(\omega) \tag{7.16}$$

In summary, signal spectra and cross-spectra serve to provide useful information about system function and disturbance. We need to estimate these spectra using sampled data. The next issue is how to estimate this spectral information using a finite number of signal samples.

It can be seen from the definitions that the computation of the spectra requires the operation of statistical expectation, which, however, needs the probability density function, which is unavailable. To circumvent this difficulty, we impose the assumption of ergodic property to the underlying stochastic processes such that the auto- and cross-covariance can be obtained by

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t-\tau)\,\mathrm{d}t \tag{7.17}$$

and

$$R_{xy}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)y(t-\tau)\,\mathrm{d}t \tag{7.18}$$

However, in practice, we are limited by a limited number of data samples. Analytical integrations of Equations 7.17 and 7.18 are precluded by the discrete time signal data. Therefore, the estimated auto- and cross-covariance functions can be computed by

$$\hat{R}_{xx}(\tau) = \frac{1}{N} \sum_{k=\tau+1}^{N} x(k)x(k-\tau) \tag{7.19}$$

and

$$\hat{R}_{xy}(\tau) = \frac{1}{N}\sum_{k=\tau+1}^{N} x(k)y(k-\tau) \tag{7.20}$$

where N is the total number of data samples, and τ is an integer. With these estimations of auto- and cross-covariance functions, the estimations of the spectral density and cross-spectral density functions are given by

$$S_{xx}(\omega) = \sum_{\tau=-N}^{N} \hat{R}_{xx}(\tau)e^{-i\omega\tau T} \tag{7.21}$$

and

$$S_{xy}(\omega) = \sum_{\tau=-N}^{N} \hat{R}_{xy}(\tau)e^{-i\omega\tau T} \tag{7.22}$$

Armed with these elementary linear systems and spectral knowledge, we are now in a position to explore the identification of an HO model.

7.4.2 Nonparametric Quasi-Linear Model

Consider the HO control system shown in Figure 7.4. The transfer functions $H(i\omega)$ and $P(i\omega)$ represent the HO model and the controlled plant model, respectively; $r(t)$ is the reference input, $e(t)$ is the tracking error, $u(t)$ is the output of the HO, and $y(t)$ is the output of the plant—all these signals are measurable—and $n(t)$ is the remnant, and $u'(t)$ is the linear component of the HO output, both of which are not measurable. We seek to identify the HO model, in the transfer function or nonparametric model form, based on measured data.

First assume, $n(t) = 0$. We can estimate the transfer function from the input and output measurements. The closed loop transfer function from $r(t)$ to $y(t)$ is

$$T(i\omega) = \frac{Y(\omega)}{R(\omega)} = \frac{H(i\omega)P(i\omega)}{1 + H(i\omega)P(i\omega)} \tag{7.23}$$

When the transfer function $P(i\omega)$ is known, we can deduce

$$H(i\omega) = \frac{T(i\omega)}{P(i\omega)(1 - T(i\omega))} \tag{7.24}$$

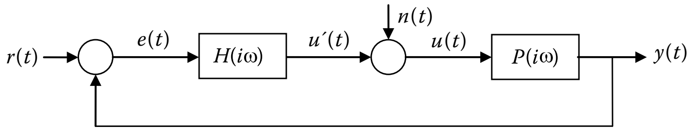


FIGURE 7.4
HO control system.

Thus, the HO model is determined. However, this type of computation might have the risk of large error when the open loop gain is high when $T(i\omega)$ approaches one. Moreover, $P(i\omega)$ must be known in this case. To avoid this risk, we can first estimate the transfer function from $r(t)$ to $u(t)$ as given by

$$S_{ur}(i\omega) = \frac{U(\omega)}{R(\omega)} = \frac{H(i\omega)}{1 + H(i\omega)P(i\omega)} \tag{7.25}$$

Then we estimate the transfer function from $r(t)$ to $e(t)$ as

$$S_{er}(i\omega) = \frac{E(\omega)}{R(\omega)} = \frac{1}{1 + H(i\omega)P(i\omega)} \tag{7.26}$$

Thus,

$$H(i\omega) = \frac{S_{ur}(i\omega)}{S_{er}(i\omega)} \tag{7.27}$$

The HO model can be obtained in this manner even if $P(i\omega)$ is unknown. But it is clear that there exists bias in the estimation when the presence of the remnant signal $n(t)$, which is stochastic noise in nature, is accounted for. Hence, we use stochastic process theory in the identification of the HO model. In particular, covariance function and spectral estimation are invoked. These operations can remove the bias in the estimation. Specifically,

$$U(\omega) = S_{ur}(i\omega)\, R(\omega) + S_{un}(i\omega)N(\omega) \tag{7.28}$$

where

$$S_{un}(i\omega) = -\frac{H(i\omega)P(i\omega)}{1 + H(i\omega)P(i\omega)} \tag{7.29}$$

Hence, considering that $n(t)$ is independent of $r(t)$, we have

$$\Phi_{ur}(\omega) = S_{ur}(i\omega)\Phi_r(\omega) \tag{7.30}$$

Similarly, $E(\omega) = S_{er}(i\omega)R(\omega) + S_{en}(i\omega)N(\omega)$ where

$$S_{en}(i\omega) = -\frac{P(i\omega)}{1 + H(i\omega)P(i\omega)} \tag{7.31}$$

Also,

$$\Phi_{er}(\omega) = S_{er}(i\omega)\Phi_r(\omega) \tag{7.32}$$

Clearly, these operations eliminate the effects of noise on the estimation of transfer function. Finally, we have

$$H(i\omega) = \frac{S_{ur}(i\omega)}{S_{er}(i\omega)} = \frac{\Phi_{ur}(\omega)}{\Phi_{er}(\omega)} \tag{7.33}$$

It follows that in a closed-loop HO system, the identification of the HO model can be done by the computation of cross-spectral density. This type of operation can eliminate the influence of noise.

7.4.3 Parametric Quasi-Linear Model

The previously presented model identification relies on the spectrum estimation and falls under nonparametric estimation. There exist some unavoidable drawbacks in this method, particularly incurred by the finite number of sampled data when used in fast Fourier transform for spectral estimation. Statistically, the estimate variance cannot decrease as the length of the sampled data is increased. Moreover, the underlying system model that is obtained through spectral estimation takes the form of discrete pairs of frequency and complex number, which are the frequency response data. This type of model cannot be used in simulation and control system synthesis. A further elaboration can be evoked by fitting a transfer function model to this frequency response data. Therefore, as an alternative, parametric model identification can overcome the foregoing weakness of a nonparametric model. The first step in parametric model identification is to specify a model structure [7]. Generally, any linear system can be described by

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \tag{7.34}$$

where $y(t)$ is the output, $u(t)$ is the control input, $e(t)$ is the white noise input, $G(q, \theta)$ is a transfer function linearly related with the control input, and $H(q, \theta)$ is another transfer function characterizing the uncertainty model, coinciding with the remnant part of the quasi-linear HO model, and θ denotes all model parameters that need to be identified. A number of choices can be made.

a. Output error model,

$$y(t) = \frac{B(q)q^{-k}}{A(q)}u(t) + e(t) \tag{7.35}$$

b. ARX model,

$$y(t) = \frac{B(q)q^{-k}}{A(q)}u(t) + \frac{1}{A(q)}e(t) \tag{7.36}$$

c. ARMAX model,

$$y(t) = \frac{B(q)q^{-k}}{A(q)}u(t) + \frac{C(q)}{A(q)}e(t) \tag{7.37}$$

where q^{-k} stands for k time delays of the input, and

$$A(q) = 1 + a_1 q^{-1} + \ldots a_{na} q^{-na} \tag{7.38}$$

$$B(q) = b_1 q^{-1} + b_2 q^{-2} + \ldots b_{nb} q^{-nb} \tag{7.39}$$

$$C(q) = 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-nc} \tag{7.40}$$

The primary task of system identification is to estimate all coefficients, which are model parameters, in the corresponding polynomials from sampled input and output data. Before using various numerical optimization methods to find the solution, we should decide first which model structure to adopt and the degree number of each polynomial in the corresponding model. A popular identification paradigm is the so-called prediction error method. Here, for each model structure, we can find a predictor $\hat{y}(t|\theta)$. Then the prediction error is

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) \tag{7.41}$$

A cost function is usually defined as

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t, \theta) \tag{7.42}$$

where N is the number of samples. The optimal estimation is $\hat{\theta}$, which minimizes this cost function, and is given by

$$\hat{\theta}_N = \arg\min V_N(\theta) \tag{7.43}$$

Suppose that the predictor $\hat{y}(t|\theta)$ is linear with respect to the parameter vector θ; specifically,

$$\hat{y}(t|\theta) = \varphi^T(t)\theta \tag{7.44}$$

Then we can have the well-known least square solution [7]:

$$\hat{\theta}_N = \left[\frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \left[\frac{1}{N} \sum_{t=1}^{N} \varphi(t) y(t) \right] \tag{7.45}$$

Thus, ARX model is the easiest one for identification because its predictor takes the form of Equation 7.44. For other model structures, pseudo-linear regression should be adopted [7].

7.4.4 Experiment and Model Identification Results

Having exposed the foregoing system identification methodologies, we proceed to a traditional manual control example to show the identification results. Here we perform a compensatory tracking experiment. The controlled plant is

$$P(s) = \frac{1}{s(Ts + 1)} \tag{7.46}$$

where $T = \frac{1}{3}$. A random reference input $r(t)$ is generated by the filter $F(s) = \frac{1}{s+2}$ driven by white noise $w(t)$. The experimental setup is shown in Figure 7.5. While observing the tracking error, the HO is required to use a joystick to force the plant output $y(t)$ to follow the random input $r(t)$.

A trained operator is used to perform the experiment in MATLAB® Simulink® platform in 150 sec. A series of signals are recorded, which are reference $r(t)$, tracking error $e(t)$, control input from joystick $u(t)$, and plant output $y(t)$ as displayed in Figure 7.6. All signals are sampled at the sampling period 0.01 sec. We now identify the HO model from $e(t)$ and $u(t)$. First, we use the spectral estimation method to identify the frequency response function

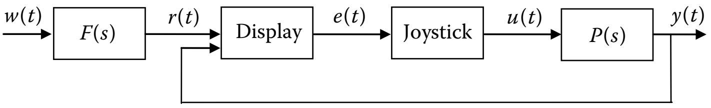


FIGURE 7.5
Experimental setup for compensatory tracking.

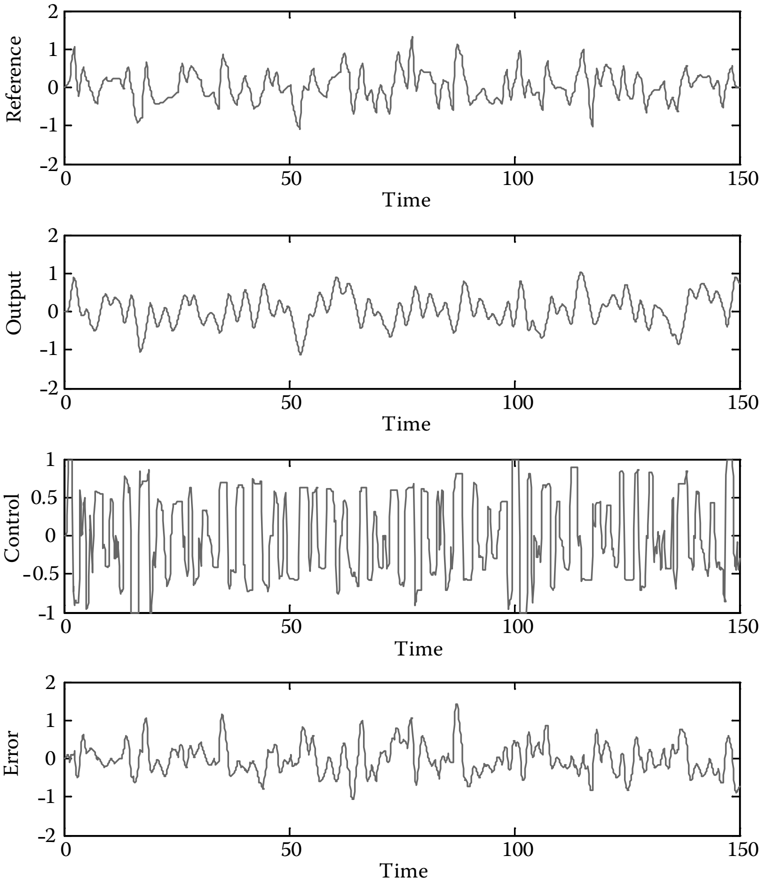


FIGURE 7.6
Experimental signals.

by using the function `spa` of the System Identification Toolbox in MATLAB. The resulting Bode diagram is shown in Figure 7.7. It is seen from this result that the frequency response function is not smooth as the usual transfer function. This is due to the finite number of sampled data. Next, we specify a parameterized model with a third-order denominator (characteristic) polynomial to identify the HO model. The identified continuous-time transfer function by using the function of `tfest` of the System Identification Toolbox is given by

$$H(s) = \frac{70.5s^2 + 518.8s + 2.432 \times 10^{-4}}{s^3 + 124.6s^2 + 641.7s + 4.205 \times 10^{-4}} \tag{7.47}$$

Its frequency response is shown in Figure 7.8. Note that this is a parametric system identification approach. The result is smoother than that from the spectral estimation method and behaves like a low-pass system, which is a common property of a physical controller.

The following concluding comments can be made. Classical system identification can be used to find an HO model in the closed-loop form. The crossover model, characterized by crossover frequency and time delay, is a widely recognized HO model. But it is only an approximation of the HO model near the crossover frequency. From the point of view of system identification, it is not an appropriate form, having two unknown parameters to fit the complex response data. Under the assumption of quasi-linear structure, both nonparametric and parametric system identification can be used to determine an HO model. However, the parametric method can result in smoother frequency response data and an analytical model, which can be used for prediction and rating.

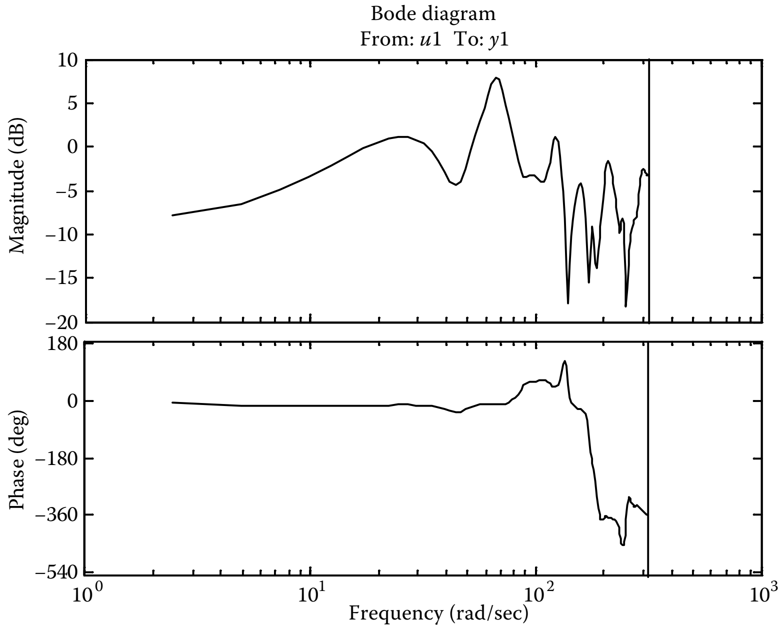


FIGURE 7.7
Bode diagram of HO model by spectral estimation method.

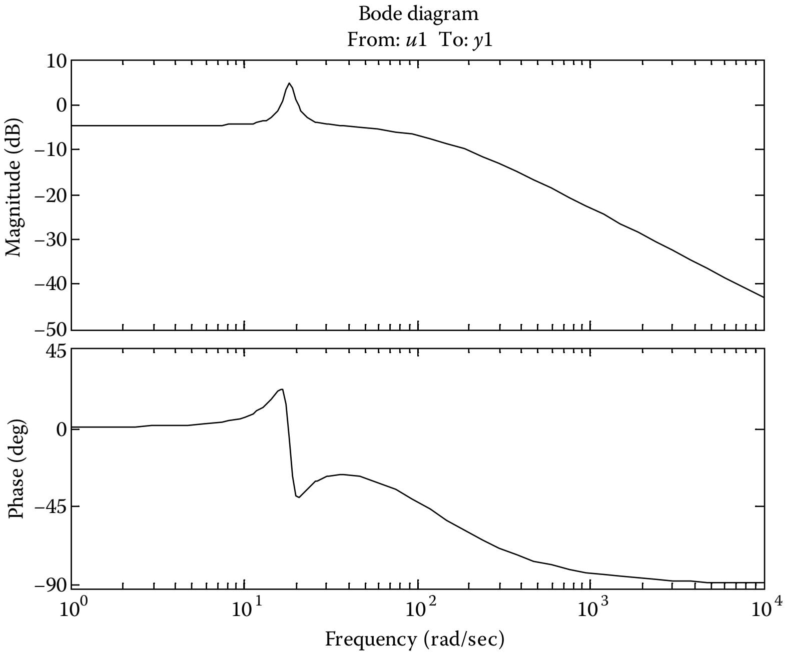


FIGURE 7.8
Frequency response for parametric HO model.

7.5 Identification through Optimal Control Theory

Frequency domain methodologies can provide us with unstructured or nonparametric models, in which the spectrum computation is the key principle. It is well known, however, that spectrum computation is strictly applicable only to stationary stochastic processes. Theoretically, an infinite number of samples are needed to compute the exact spectral components. But the finite number of samples incurs the problem of spectral leakage. Moreover, a frequency response function characterizes only the steady-state behavior, and the transient behavior is lost from the signal (even though transient response to a specific input and initial conditions can be determined once a frequency-domain model is identified). In the parametric model method, prespecification of a model structure is crucial for the identification process. But a general guideline is still not available. These flaws of transfer function estimation methods lead us to consider the time-domain modeling approach as an alternative. In time domain modeling, modern control theory is intensively employed. Two prominent features can characterize the modern control theory. First, resorting to the state space formulation, the controller and the plant are highly structured. A model-based controller consists of state feedback and state estimation. Second, optimization-based methods greatly facilitate the controller design. When a control problem is converted into a mathematical optimization problem, systematic steps can be taken to find the analytical or numerical solutions. Optimal control theory is such a paradigm. From optimal control theory, we can obtain a series of functional modalities, which constitute the overall closed-loop system. Static state feedback, observer and Kalman filter are such examples. Meanwhile, Lyapunov and Riccati equations are indispensable tools to implement the computations necessary to find these modalities.

Amazingly, the optimal control theory as outlined before is applicable in HO modeling. Kleinman, Baron, and Levinson [9] are pioneers of appreciating it in HO modeling. They put forth the hypothesis that an HO can take actions to mimic the optimal control law. The present section explores this problem in detail. We first present the elementary linear regulator problem.

7.5.1 Linear Regulator Problem

Modern theory of control and estimation is widely employed in many engineering areas. Its foundation rests on the state space theory. Within this framework, multivariable inputs and outputs can be tackled uniformly. Among numerous controller design methods, we adopt here the observer-based controller owing to its prominent structural properties and systematic design procedures. To be specific, the controller consists of static state feedback and a dynamic state estimator.

The first step of optimal control theory is to define a performance measure to transform some engineering objectives to a mathematical optimization problem. Minimal energy, minimal time, etc., are such formulated problems. The regulator problem is specialized to find a control law that drives the system state from its initial condition to zero while minimizing a quadratic form of cost function defined as the performance measure. Under these circumstances, the optimal control is a static state feedback and is obtained by the solution of a stationary Riccati equation. If the state variable is not directly measurable, we need a Kalman filter to estimate the state variable. Hence, another Riccati equation is needed to generate the optimal estimator gain. As an application of a well-known separation principle [10], this state estimate can be used in lieu of real state feedback. All preceding procedures constitute the standard linear regulator theory. When processes and measurement noises are considered and obey the white Gaussian assumption, it is also termed linear quadratic Gaussian (LQG) control.

For an HO controller, there exists a time delay in observation due to the inherent physiological limitations. This might prevent us from directly using LQG theory. To compensate for the time delay, a state predictor is need as an extra modality in the HO model. The ultimate controller is the cascade of a Kalman filter, a least mean square predictor, and a feedback gain. Next we review the LQG theory without and with time delay, respectively. The whole derivation is extracted mainly from [10–13].

7.5.2 LQG Controller without Time Delay

The underlying system is considered to be a linear, time-invariant plant. So it can be described by a state space model as

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \tag{7.48}$$

with the output equation

$$y(t) = Cx(t) + v(t) \tag{7.49}$$

where $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ is the control input vector, $w(t) \in R^n$ is the process noise vector, $y(t) \in R^p$ is the output vector (in manual control, the observed output is usually the error and the error rate), and $v(t) \in R^p$ is the measurement noise vector, which accounts for the attention allocation and perception thresholds for the HO. Here $A \in R^{n \times n}$, $B \in R^{n \times m}$, and $C \in R^{p \times n}$ are constant coefficient matrices with the indicated dimensions. We assume that all noise vectors are white with zero mean and

$$E[w(t_1)w^T(t_2)] = W\delta(t_1 - t_2),\ E[v(t_1)v^T(t_2)] = V\delta(t_1 - t_2) \tag{7.50}$$

where the variance matrices W and V are both positive definite and express the intensity of the noise, and $\delta(t)$ is the Kronecker delta operator, $\delta(0) = 1$, $\delta(t) = 0$ for $t \neq 0$. Under the condition of controllability and observability [10], the linear regulator problem is posed as finding a state feedback $u = -Kx$ such that the quadratic performance measure

$$J(u) = E\left\{\lim_{T\to\infty} \frac{1}{T}\int_0^T (x^T Q x + u^T R u)\,\mathrm{d}t\right\} \tag{7.51}$$

is minimized. Here $E\{\cdot\}$ is the expectation operator, Q is a non-negative weighting matrix such that $(A, Q^{1/2})$ is completely observable, and R a positive definite weighting matrix. This is the standard infinite-time regulator problem [10], whose solution is given by

$$u^*(t) = -Kx = -R^{-1}B^T Px \tag{7.52}$$

where P is the solution of the following stationary Riccati equation:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \tag{7.53}$$

There are two results associated with this optimal control strategy. The first is that the closed-loop system is stable. Specifically,

$$\dot{x} = (A - BK)x = (A - BR^{-1}B^T P)x \tag{7.54}$$

is asymptotically stable. The second is the optimal cost $J(u^*(t))$, which is given by

$$J(u^*) = \mathrm{tr}(PW) \tag{7.55}$$

where tr denotes the trace of the matrix product of P and W, which is meaningful in rating the performance of the optimal controller.

The optimal control law is a static feedback. When the state variable is not directly measurable, we can apply the well-known certainty equivalence principle [10], namely, replace $x(t)$ by its optimal estimation $\hat{x}(t)$, thereby leading to a dynamic controller. The state estimator can be formulated as [10]

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \tag{7.56}$$

where $\hat{x}(t)$ is the estimate of $x(t)$, and L is a gain matrix to be designed. It can be derived in two different ways. One of them is posed as observer design, which requires that the eigenvalues of $A - LC$ have negative real parts. The other is the Kalman filter design [10], which is adopted here. In the Kalman filter setting, a statistical criterion is sought to minimize the mean squared estimation error of $x(t)$, namely

$$E\{[x(t) - \hat{x}(t)][x(t) - \hat{x}(t)]^T\} \tag{7.57}$$

The solution is given by

$$L = -\Sigma C^T V^{-1} \tag{7.58}$$

where Σ satisfies the following Riccati equation:

$$A\Sigma + \Sigma A^T + W - \Sigma C^T V^{-1} C\Sigma = 0 \tag{7.59}$$

The final controller is a dynamical system given by

$$\dot{\hat{x}} = (A - BK - LC)\hat{x} + Ly \tag{7.60}$$

$$u = -K\hat{x} \tag{7.61}$$

This is the normative observer-based controller. The structure of the resultant controller is closely related with controlled plant dynamics. The parameter matrix in the controller is determined by the state feedback gain K and the Kalman filter gain L, both of which are obtained by two optimization procedures and two Riccati equations. Next we investigate the case of time delay.

7.5.3 LQG Controller with Time Delay

To scan the output signal by an HO, transport time and time of signal processing by the central nervous system are needed. Hence, time delay is unavoidable to perceive the output signals. Thus instead of Equation 7.49, an HO actually receives the observation

$$y(t) = Cx(t - \tau) + v(t - \tau) \tag{7.62}$$

where τ denotes the time delay. The problem now becomes minimization of the quadratic performance measure Equation 7.51 under the delayed observation Equation 7.62. We should first attempt to solve the prediction problem by means of the least mean square criterion [12], that is,

$$\hat{x}(t) = E\left\{x(t)\middle|y(\sigma), \sigma \le t\right\} \tag{7.63}$$

which is a conditional expectation of $x(t)$ on $y(\sigma)$, $\sigma \le t$. Let us rewrite the dynamic equation with explicit time instant at $t - \tau$,

$$\dot{x}(t - \tau) = Ax(t - \tau) + Bu(t - \tau) + w(t - \tau) \tag{7.64}$$

The evolution of the underlying state variables is driven by two contributors, the deterministic input $u(t-\tau)$ and the stochastic input $w(t-\tau)$. Linear system property enables us to decompose $x(t-\tau)$ as the sum of $x_u(t)$ and $x_w(t)$, where

$$\dot{x}_u(t) = Ax_u(t) + Bu(t-\tau) \tag{7.65}$$

and

$$\dot{x}_w(t) = Ax_w(t) + w(t-\tau) \tag{7.66}$$

Because $x_u(t)$ is totally deterministic, it can be calculated from the control input $u(t)$. Therefore, the estimation of $x(t)$ can be determined as

$$\hat{x}(t) = x_u(t+\tau) + \hat{x}_w(t+\tau) \tag{7.67}$$

where $\hat{x}_w(t+\tau)$ denotes the least mean square estimation of $x_w(t+\tau)$. Because of the whiteness of $w(t)$, this estimate is given by

$$\hat{x}_w(t+\tau) = e^{A\tau}x_w(t) \tag{7.68}$$

We have

$$\hat{x}(t) = x_u(t+\tau) + e^{A\tau}x_w(t) \tag{7.69}$$

and

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + e^{A\tau}w(t-\tau) \tag{7.70}$$

Let us examine the state estimation error, given by

$$x_e(t) = x(t) - \hat{x}(t) = x_w(t+\tau) - \hat{x}_w(t+\tau) = x_w(t+\tau) - e^{A\tau}x_w(t) \tag{7.71}$$

But by virtue of Equation 7.66, we have

$$x_w(t+\tau) = e^{A\tau}x_w(t) + \int_t^{t+\tau} e^{A(t+\tau-\sigma)}w(\sigma-\tau)\,\mathrm{d}\sigma \tag{7.72}$$

Therefore,

$$x_e(t) = \int_t^{t+\tau} e^{A(t+\tau-\sigma)}w(\sigma-\tau)\,\mathrm{d}\sigma \tag{7.73}$$

Because

$$x(t) = \hat{x}(t) + x_e(t) \tag{7.74}$$

and $\hat{x}(t)$ is independent of $x_e(t)$, we have

$$E\{x^T(t)Qx(t)\} = E\{\hat{x}^T(t)Q\hat{x}(t)\} + E\left\{x_e^T(t)Qx_e(t)\right\} \tag{7.75}$$

Hence, the performance (Equation 7.51) can be rewritten as

$$J(u) = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T \left(x_e^T Q x_e + \hat{x}^T Q\hat{x} + u^T R u\right)\mathrm{d}t\right\} \tag{7.76}$$

But the first term is independent of the control $u(t)$. Hence, equivalently, we can optimize

$$J_1(u) = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T (\hat{x}^T Q\hat{x} + u^T R u)\,\mathrm{d}t\right\} \tag{7.77}$$

subject to the constraints

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + e^{A\tau}w(t-\tau) \tag{7.78}$$

It is clear that the optimal control is still given by

$$u^*(t) = -K\hat{x}(t) \tag{7.79}$$

where K remains the same as in Equation 7.52. This result indicates that the time delay has no effect on the optimal control solution. But we cannot obtain the state estimation from Equation 7.78 because of the unknown noise term. However we can give an intuitive and concise result, and the rigorous derivation is given in [12]. The ultimate result is the same. We first make use of the Kalman filter to attain the state estimate $\hat{x}(t-\tau)$, namely

$$\dot{\hat{x}}(t-\tau) = A\hat{x}(t-\tau) + Bu(t-\tau) + L(y(t) - C\hat{x}(t-\tau)) \tag{7.80}$$

where

$$L = -\Sigma C^T V^{-1} \tag{7.81}$$

$$A\Sigma + \Sigma A^T + W - \Sigma C^T V^{-1} C\Sigma = 0 \tag{7.82}$$

Having the state estimate $\hat{x}(t-\tau)$, the state is evolved by

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \tag{7.83}$$

If we assign the initial state of the above system as $\hat{x}(t-\tau)$ at time $t-\tau$, the best that we can do is to regard the state estimate at time t, $\hat{x}(t)$, as the superposition of the zero input response and the zero state response, which follows from the principle of superposition in linear system theory. In other words, we ignore the effect of noise after the time $t-\tau$, thus

$$\hat{x}(t) = e^{A\tau}\hat{x}(t-\tau) + \int_{t-\tau}^{t} e^{A(t-s)}Bu(s)\,\mathrm{d}s \tag{7.84}$$

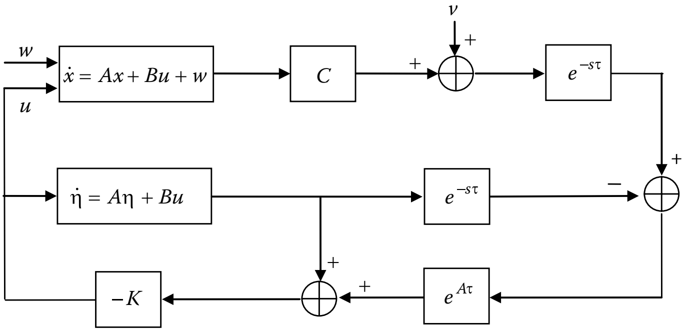


FIGURE 7.9
Optimal controller with time delay.

Define

$$\dot{\eta}(t) = A\eta(t) + Bu(t) \tag{7.85}$$

Then

$$\eta(t) = e^{A\tau}\eta(t-\tau) + \int_{t-\tau}^{t} e^{A(t-s)}Bu(s)\,\mathrm{d}s \tag{7.86}$$

Therefore,

$$\hat{x}(t) = e^{A\tau}\hat{x}(t-\tau) + \eta(t) - e^{A\tau}\eta(t-\tau) = \eta(t) + e^{A\tau}(\hat{x}(t-\tau) - \eta(t-\tau)) \tag{7.87}$$

The overall system is represented in Figure 7.9.

With this foundation, we next treat the HO model through optimal control theory.

7.5.4 Optimal Control Model for the Human Operator

An HO has self-adaptive and well-motivated abilities while being subject to inherent limitations by virtue of the psycho-physiological character. When the control objectives and the knowledge of the plant are captured in some form by HOs, their actions have the tendency to be optimal in some sense. The plausibility of this conjecture has been validated by many researchers. In this section, we illustrate in detail how optimal control theory, specifically linear regulator theory, can be used to model the HO.

We first introduce the controlled plant, which has a linear, time-invariant state space model with sensor and actuator dynamics, as given by

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \tag{7.88}$$

where $x \in R^n$ is the vector of system states, $u \in R^m$ is the vector of control inputs, A and B are constant system matrices, and $w \in R^n$ is the noise vector for independent disturbance with assumed statistics:

$$E\{w(t)\} = 0,\ E\{w(t_1)w^T(t_2)\} = W\delta(t_1 - t_2) \tag{7.89}$$

The outputs are a linear combination of the state variables:

$$y(t) = Cx(t) \tag{7.90}$$

where $y \in R^p$ is the vector of system outputs that can be displayed for the HO to observe, sometimes with perceived variables included, and C is a constant matrix with appropriate dimension.

When a human is in the loop as a controller, the physiological system of the operator will interact with the physical system of the controlled plant. Considering the underlying control task and inherent psycho-physical limitations of the operator, we can visualize the overall HO as three cascaded blocks with different functions. They are time delay, neuromuscular lag, and equalization network, as described below.

1. Time delay: An HO perceives the output through visual scanning and sensing. Furthermore, central signal processing in the nervous system needs time to perform. A time delay τ is introduced due to these effects. Typically $\tau = 0.2$ sec [9]. There exist also some random effects in human response. Therefore the observed outputs can be represented as

   $$y_o = Cx(t - \tau) + v_y(t - \tau) \tag{7.91}$$

   where the observer noise can be statistically described as

   $$E\{v_y\} = 0, E\left\{v_y(t_1)v_y^T(t_2)\right\} = V_y\delta(t_1 - t_2) \tag{7.92}$$

2. Equalization network: This block represents an internal model corresponding to the controlled plant dynamics and the control task by which the HO can optimize the control strategy. It is the central part of the HO model.
3. Neuromuscular dynamics and motor noise: HOs exercise control through their limbs. The limitations of the control reaction time and speed can be lumped into the neuromotor lag effect, which can be described by a first-order dynamic model [4]:

   $$H_n(s) = \frac{1}{\tau_N s + 1} \tag{7.93}$$

   where τ_N is a time constant reflecting the lag effect. Typically $\tau_N = 0.1$ sec. Its time domain input and output relationship is

   $$\tau_N \dot{u} + u = u_c \tag{7.94}$$

   where u_c, the output of the equalization network, is the input to the $H_n(s)$, and u is the output of $H_n(s)$. When considering motor noise, we have

   $$\dot{u} = -\frac{1}{\tau_N}u + \frac{1}{\tau_N}u_c + \frac{1}{\tau_N}v_m \tag{7.95}$$

   where u is the actual control applied to the plant, and v_m represents motor noise with zero mean and intensity V_m.

A schematic diagram for the HO system is shown in Figure 7.10.

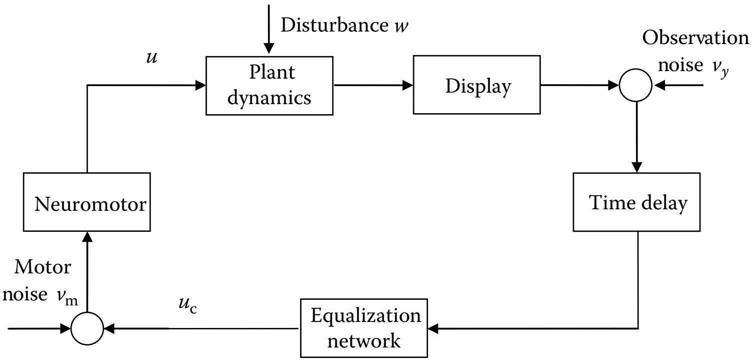


FIGURE 7.10
HO system diagram.

7.5.5 Human Optimal Control Model (OCM)

Through a series of sensing, learning, and actuation, it can be expected that HOs become well trained and skillful. They can implicitly utilize system dynamics to make actions and predictions. That is to say that the underlying dynamic model of the controlled plant is of an internal model to the HOs. Hence, we can regard that their behavior tends to be nearly optimal in some sense. A reasonable conjecture can be made to regard that the HO executes control actions by complying with some optimal control paradigm. To be specific, let us associate HO control with the LQG problem. Here, for simplicity, we only consider the single input case, but the extension to the multiple input case is straightforward. We define the performance measure

$$J(u) = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T (x^TQx + ru^2 + g\dot{u}^2)\mathrm{d}t\right\} \tag{7.96}$$

where Q is a non-negative definite matrix as before, and $r \geq 0$ and $g \geq 0$ are scalars. A well-trained operator attempts to minimize this performance measure. The inclusion of $\dot{u}$ in the cost function is due to human limitations on the action rate. Its subtle relationship with neuromuscular dynamics will be revealed later. This type of performance resembles the conventional compensatory tracking task in which manual control is required to minimize the mean square error. This cost function is different from Equation 7.51. In order to apply the standard linear regulator results, we need some modifications. Define an augmented state variable vector as

$$x_u = [x^T\ u^T]^T \tag{7.97}$$

Then we can find the state equation

$$\dot{x}_u = A_u x_u + b_0\dot{u} + w_0 \tag{7.98}$$

where

$$A_u = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix},\ b_0 = \begin{bmatrix} 0_{n\times1} \\ 1 \end{bmatrix},\ w_0 = \begin{bmatrix} w \\ 0 \end{bmatrix} \tag{7.99}$$

The performance measure can be rewritten as

$$J(u) = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T \left(x_u^T Q_u x_u + g\dot{u}^2\right)\mathrm{d}t\right\} \tag{7.100}$$

where

$$Q_u = \begin{bmatrix} Q_{n\times n} & 0_{n\times 1} \\ 0_{1\times n} & r \end{bmatrix} \tag{7.101}$$

Then, according to Equation 7.52, the optimal state feedback is given by

$$\dot{u}^* = -\frac{1}{g} b_0^T P_u x_u = [-L_x \quad -l_u]\begin{bmatrix} x \\ u^* \end{bmatrix} \tag{7.102}$$

where L_x and l_u are appropriate matrix partitions corresponding to the state variable partitions, and P_u satisfies the Riccati equation

$$P_u A_0 + A_0^T P_u + Q_u - \frac{1}{g} P_u b_0 b_0^T P_u = 0 \tag{7.103}$$

We now obtain the explicit controller equation

$$\dot{u}^* = -l_u u^* - L_x x \tag{7.104}$$

Comparing with neuromotor dynamics, we discover that if we can specify the weight g such that $\tau_N = \frac{1}{l_u}$ and include the motor noise $\frac{1}{\tau_N} v_{\mathrm{m}}$ into the above expression, we arrive at

$$\dot{u} = -\frac{1}{\tau_N} u + \frac{1}{\tau_N} u_{\mathrm{c}} + \frac{1}{\tau_N} v_{\mathrm{m}} \tag{7.105}$$

where

$$u_{\mathrm{c}} = -L_x x \tag{7.106}$$

This result reveals the mechanism, as we claimed before, of capturing the neuromotor dynamics by introducing the control rate term in the performance measure. Up to now, we have assumed that the state variables are available. The more practical situation is one in which they are not available. Then they should be replaced by their estimates $\hat{x}(t)$, which are obtained from the Kalman filter and predictor. From the state Equation 7.88 and the output Equations 7.91, we can design a state estimator as

$$\dot{\hat{x}}(t-\tau) = A\hat{x}(t-\tau) + Bu(t-\tau) + \Sigma C^T V^{-1}(y_o - C\hat{x}(t-\tau)) \tag{7.107}$$

where Σ is the solution of the following stationary Riccati equation:

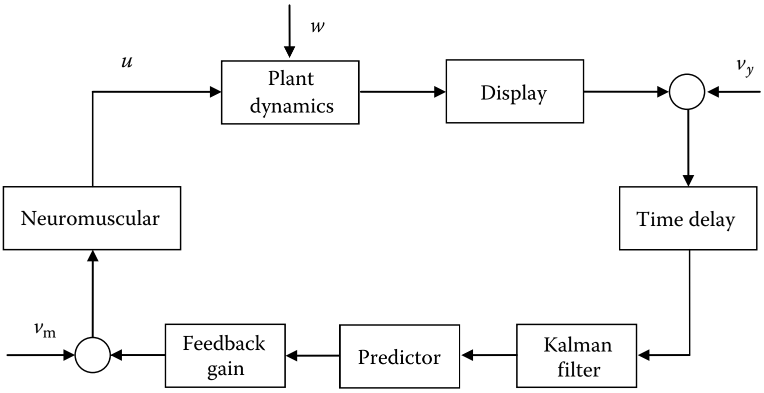


FIGURE 7.11
Optimal control model.

$$A\Sigma + \Sigma A^T + W - \Sigma C^T V^{-1} C\Sigma = 0 \tag{7.108}$$

The overall control structure is depicted in Figure 7.11. We notice that the motor noise is added after the feedback gain is completed. We need to investigate further the effect of this addition of the noise term.

7.5.6 Motor Noise Effect

When motor noise is taken into consideration, the control equation is

$$\tau_N \dot{u} + u = u_c + v_m \tag{7.109}$$

Generally, the intensity of v_m, V_m, is much smaller than $E\{u_c^2\}$. The influence of v_m on the control performance is negligible. However, its effect on the state estimation may be significant. Hence, the Kalman filter should be modified. Substituting Equation 7.105 into Equation 7.98, we obtain

$$\dot{x}_u = A_u' x_u + b_1 u_c + w_1 \tag{7.110}$$

$$y_o = C_u x_u(t - \tau) + v(t - \tau) \tag{7.111}$$

where

$$A_u' = \begin{bmatrix} A & B \\ 0 & -\dfrac{1}{\tau_N} \end{bmatrix}, \; b_1 = \begin{bmatrix} 0_{n\times 1} \\ \dfrac{1}{\tau_N} \end{bmatrix}, \; w_1 = \begin{bmatrix} w \\ \dfrac{v_m}{\tau_N} \end{bmatrix}, \; C_u = [C \quad 0], \; E\{w_1 w_1^T\} = \begin{bmatrix} W & 0 \\ 0 & \dfrac{V_m}{\tau_N^2} \end{bmatrix} = W_1.$$

The dynamic equation for state estimation is

$$\dot{\hat{x}}_u(t-\tau) = A_u' \hat{x}_u(t-\tau) + b_1 u_c(t-\tau) + \Sigma_u C^T V^{-1}(y_o - C_u \hat{x}_u(t-\tau)) \tag{7.112}$$

where Σ_u is the solution of

$$A'_u\Sigma_u + \Sigma_u A'^T_u + W_1 - \Sigma_u C^T_u V^{-1} C_u \Sigma_u = 0 \tag{7.113}$$

As proven in Equation 7.87, $\hat{x}_u(t)$ is given by

$$\hat{x}_u(t) = \eta(t) + e^{A'_u\tau}(\hat{x}_u(t-\tau) - \eta(t-\tau)) \tag{7.114}$$

and

$$\dot{\eta}(t) = A'_u\eta(t) + b_1 u_c(t) \tag{7.115}$$

Now, it can be concluded that the overall OCM is composed of the cascade of Kalman filter, least mean predictor, and the optimal feedback gain. There are two different plants for the synthesis of state feedback gain and Kalman filter gain, which are very different from the usual observer-based output feedback control. We observe that time delay has substantially complicated the OCM model. The modified optimal control model is developed to alleviate the effect of time delay.

7.5.7 Modified Optimal Control Model (MOCM)

OCM is fairly comprehensive. State feedback, state estimation, and prediction should all be implemented to synthesize an HO model. Unlike the OCM, the MOCM developed by [14] can be used to derive the transfer function in pole-zero form, which is input compatible with the OCM.

The underlying plant of SISO to be controlled and observed is

$$\dot{x} = Ax + Bu + w \tag{7.116}$$

$$y_o = Cx + v_y \tag{7.117}$$

where the system state $x \in R^n$ is an augmentation of the plant dynamics and the noise dynamics. A notable difference with OCM is that time delay is no longer present in the observation. The time delay block is moved to the position between the plant and neuromuscular lag blocks as shown in Figure 7.12.

The most significant change here compared with the OCM is the elimination of the predictor due to the movement of the time delay block. This arrangement makes state estimation easier to implement than the case in which time delay is present in the observation. Another reason for doing this is to consider the case of number of operator inputs greater than outputs, when moving the delay to the output will result in a low order representation [14]. The pure time delay in the transfer function form is $e^{-\tau s}$. It can be approximated by the Pade approximation, a rational transfer function. We can choose the order of the approximate transfer function base on the considered frequency range. For HO modeling, a second-order Pade approximation is sufficient. A second-order transfer function approximation [14] is given by

$$\frac{U(s)}{U_p(s)} = \frac{1 - \dfrac{\tau s}{2} + \dfrac{\tau^2 s^2}{12}}{1 + \dfrac{\tau s}{2} + \dfrac{\tau^2 s^2}{12}} \tag{7.118}$$

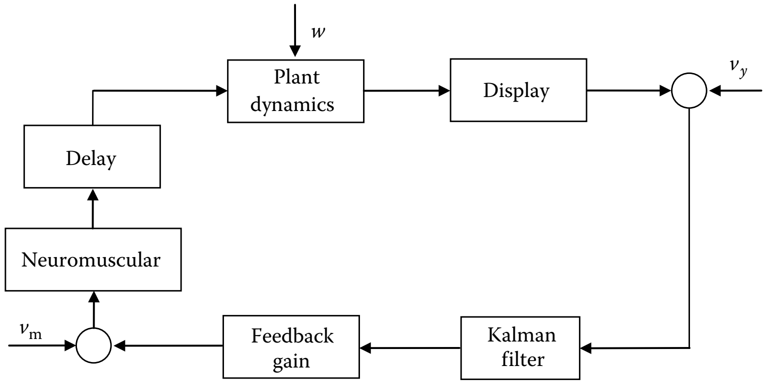


FIGURE 7.12
Modified optimal control model.

This can be transformed into the state space form as

$$\dot{x}_d = A_d x_d + B_d u_p \tag{7.119}$$

$$u = C_d x_d + u_p \tag{7.120}$$

where $x_d \in R^2$ denotes the Pade delay state variable, a two-dimension vector in this case, u_p is the input, u is the output. All coefficient matrices are given by

$$A_d = \begin{bmatrix} -\dfrac{6}{\tau} & -\dfrac{1}{\tau} \\ \dfrac{12}{\tau} & 0 \end{bmatrix}, \; B_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; C_d = \begin{bmatrix} -\dfrac{12}{\tau} & 0 \end{bmatrix} \tag{7.121}$$

In MOCM, the dynamical system of Pade approximation is augmented to plant dynamics. So we augment the vector of state variable to $x_m = \begin{bmatrix} x^T & x_d^T \end{bmatrix}^T \in R^{n+2}$. Then a newly augmented dynamical system is obtained as

$$\dot{x}_m = A_m x_m + B_m u_p + w_m \tag{7.122}$$

$$y_o = C_m x_m + v_y \tag{7.123}$$

where

$$A_m = \begin{bmatrix} A & BC_d \\ 0_{2\times n} & A_d \end{bmatrix}, \; B_m = \begin{bmatrix} B \\ B_d \end{bmatrix}, \; C_m = [C \quad 0_{1\times 2}], \; w_m = \begin{bmatrix} w \\ 0_{2\times 1} \end{bmatrix}$$

Consider the performance measure to be minimized by the HO:

$$J(u_p) = E\left\{ \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(y_o^T Q_y y_o + r u_p^2 + g \dot{u}_p^2 \right) \mathrm{d}t \right\} \tag{7.124}$$

where Q_y is a non-negative weighting matrix, and $r \geq 0$ and $g > 0$ are scalar weights for u_p and $\dot{u}_p$, respectively. Here y_o, rather than x_m, is punished as the regulated variable, and it makes no essential difference with the previous treatment. In order to apply standard linear regulator results, we define a further augmented state variable $x_{mo} = \begin{bmatrix} x_m^T & u_p^T \end{bmatrix}^T$, to obtain

$$\dot{x}_{mo} = A_{mo} x_{mo} + B_{mo} \dot{u}_p + w_{mo} \tag{7.125}$$

$$y_o = C_{mo} x_{mo} + v_y \tag{7.126}$$

where

$$A_{mo} = \begin{bmatrix} A_m & B_m \\ 0_{1\times(n+2)} & 0 \end{bmatrix},\ B_{mo} = \begin{bmatrix} 0_{(n+2)\times 1} \\ 1 \end{bmatrix},\ C_{mo} = [\ C_m \quad 0\ ],\ w_{mo} = \begin{bmatrix} w_m \\ 0 \end{bmatrix}$$

The performance measure may be rewritten as

$$J(u_p) = E\left\{ \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(x_{mo}^T Q_m x_{mo} + g \dot{u}_p^2 \right) \mathrm{d}t \right\} \tag{7.127}$$

where

$$Q_m = \begin{bmatrix} C_{mo}^T Q_y C_{mo} & 0 \\ 0 & r \end{bmatrix} \tag{7.128}$$

We now have a standard linear regulator problem, which has the optimal solution

$$\dot{u}_p^* = -\frac{1}{g} B_{mo}^T P_{mo} x_{mo} = -[\ L_{mox} \quad l_{mou}\ ] \begin{bmatrix} x_m \\ u_p^* \end{bmatrix} \tag{7.129}$$

where P_{mo} is the solution of

$$P_{mo} A_{mo} + A_{mo}^T P_{mo} + Q_m - \frac{1}{g} P_{mo} B_{mo} B_{mo}^T P_{mo} = 0 \tag{7.130}$$

L_{mox} (vector) and l_{mou} (scalar) are the partitions of the matrix $-\frac{1}{g}B_{\text{mo}}^T P_{\text{mo}}$ corresponding to the partition of the state variable x_{mo}. Thus,

$$\dot{u}_p^* = -l_{\text{mou}} u_p^* - L_{\text{mox}} x_{\text{m}} \tag{7.131}$$

or

$$\tau_N \dot{u}_p^* + u_p^* = u_{\text{c}},\ \tau_N = \frac{1}{l_{\text{mou}}},\ u_{\text{c}} = -L_{\text{m}} x_{\text{m}} = -\frac{1}{l_{\text{mou}}} L_{\text{mox}} x_{\text{m}} \tag{7.132}$$

Taking motor noise into account yields

$$\tau_N \dot{u}_p + u_p = u_{\text{c}} + v_{\text{m}} \text{ or } \dot{u}_p = -\frac{1}{\tau_N} u_p + \frac{1}{\tau_N} u_{\text{c}} + \frac{1}{\tau_N} v_{\text{m}} \tag{7.133}$$

where v_{m} stands for motor noise. Its statistics are

$$E\{v_{\text{m}}\} = 0,\ E\{v_{\text{m}}(t_1)v_{\text{m}}(t_2)\} = V_{\text{m}}\delta(t_1 - t_2) \tag{7.134}$$

This noise term has an effect on state estimation. We need to modify the dynamic equations in order to enable the Kalman filter to accommodate this noise term. Substituting $\dot{u}_p$ in Equation 7.133 into Equation 7.125 gives

$$\begin{bmatrix} \dot{x}_{\text{m}} \\ \dot{u}_p \end{bmatrix} = \begin{bmatrix} A_{\text{m}} & B_{\text{m}} \\ 0_{1\times(n+2)} & 0 \end{bmatrix} \begin{bmatrix} x_{\text{m}} \\ u_p \end{bmatrix} + \begin{bmatrix} 0_{(n+2)\times 1} \\ 1 \end{bmatrix} \left(-\frac{1}{\tau_N} u_p - \frac{1}{\tau_N} u_{\text{c}} + \frac{1}{\tau_N} v_{\text{m}} \right) + w_{\text{mo}} \tag{7.135}$$

This can be arranged as

$$\dot{x}_{\text{mo}} = \tilde{A}_{\text{mo}} x_{\text{mo}} + \tilde{B}_{\text{mo}} u_{\text{c}} + E\tilde{w} \tag{7.136}$$

where

$$\tilde{A}_{\text{mo}} = \begin{bmatrix} A_{\text{m}} & B_{\text{m}} \\ 0_{1\times(n+2)} & -\frac{1}{\tau_N} \end{bmatrix},\ \tilde{B}_{\text{mo}} = \begin{bmatrix} 0_{(n+2)\times 1} \\ -\frac{1}{\tau_N} \end{bmatrix},\ E = \begin{bmatrix} I & 0_{(n+2)\times 1} \\ 0 & \frac{1}{\tau_N} \end{bmatrix},\ \tilde{w} = \begin{bmatrix} w_{\text{mo}} \\ v_{\text{m}} \end{bmatrix}$$

Note that even though there is no change in the definition of the state variable, the state equation is changed due to the introduction of motor noise. The output equation is kept the same. Now we can employ Kalman filter to fulfill the state estimation, which is given by

$$\dot{\hat{x}}_{\text{mo}} = \tilde{A}_{\text{mo}} \hat{x}_{\text{mo}} + \tilde{B}_{\text{mo}} u_{\text{c}} + L(y_{\text{o}} - C_{\text{mo}} - \hat{x}_{\text{mo}}) \tag{7.137}$$

Here L is the Kalman filter gain, given by

$$L = \Sigma_1 C_{\text{mo}}^T V_y^{-1} \tag{7.138}$$

where Σ_1 is the solution of

$$\tilde{A}_{\text{mo}}\Sigma_1 + \Sigma_1 \tilde{A}_{\text{mo}}^T + W_1 - \Sigma_1 C_{\text{mo}}^T V_y^{-1} \Sigma_1 = 0 \tag{7.139}$$

and

$$W_1 = \begin{bmatrix} W & 0 \\ 0 & V_{\text{m}} \end{bmatrix} \tag{7.140}$$

Thus, we have obtained all necessary elements that constitute the final controller. But many interactive procedures are needed to tune all the parameters to fit the measured input and output data. The tuned parameters include the weighting matrices in the performance measure, intensity of the processes, and the observation noise. This is a generally laborious and difficult process. It usually takes several rounds of trials to reach a satisfactory result. In conclusion, OCM and MOCM are not system identification oriented. We next explore the optimal control model by means of system identification based on measured data.

7.5.8 Identification of Optimal Control Model

An HO is characterized as an optimal controller in structure. The underlying parameter in the model should be determined to fit the input and output data. A data-based method to identify the parameters in the optimal control model is highly advocated so that we are not required to blindly tune some parameters. Under the framework of OCM, the following parameters should be identified:

1. Time delay τ
2. Observation noise intensity V_y
3. Motor noise intensity V_{m}
4. Optimal gain vector

System identification is a powerful tool for many engineering applications. Much work has been carried out in this area [7]. Model selection, parameter estimation, and validation constitute the main three stages in dynamic system identification. Identifiability is an important issue in system identification. It concerns the problem of uniqueness of parameter estimation. We can tackle identifiability from both deterministic and stochastic points of view. We now take an example to illustrate the problem of identifiability [15].

A single input and single output second-order system is given by

$$\dot{x} = Ax + Bu = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} x + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u \tag{7.141}$$

$$y = Cx = [c_1 \quad c_2]x \tag{7.142}$$

In the state space model, all the model parameters lie in the A, B, and C matrices with a total number of eight. However, with the same system, the transfer function form for control u to output y is

$$G(s) = C(sI - A)^{-1}B = \frac{k_1 s + k_2}{s^3 - k_3 s - k_4} \tag{7.143}$$

where $k_1 = c_1 b_1 + c_2 b_2$, $k_2 = c_2 a_{12} b_2 - c_1 a_{22} b_1 + c_2 a_{21} b_1 - c_2 a_{11} b_2$ and $k_2 = a_{11} + a_{22}$, $k_4 = a_{12} a_{21} - a_{11} a_{22}$.

From the identifiability point of view, there is no one-to-one correspondence between the parameters in the state space model and the parameters in the transfer function model. Therefore, the general state space model may be unidentifiable. In order to realize one-to-one mapping, we can use various canonical forms for the liner system [7]. One type of canonical form is

$$\dot{x} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ -a_1 & -a_2 & \cdots & -a_n \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u \tag{7.144}$$

$$y = [c_1 \quad c_2 \quad \cdots \quad c_n]x \tag{7.145}$$

It can be proven that the transfer function from u to y is given by

$$G(s) = \frac{c_1 s^{n-1} + c_2 s + \cdots c_n}{s^n + a_1 s^{n-1} + \cdots + a_n} \tag{7.146}$$

It is now clear that there exists one-to-one correspondence between parameters in the canonical form and transfer function models.

In stochastic state space model identification, the innovation model plays an important role. Consider the linear stochastic dynamical system

$$\dot{x} = Ax + w \tag{7.147}$$

$$y = Cx + v \tag{7.148}$$

where x represents the system state, y is the output, and w and v are process and output zero mean white noises with covariances W and V, respectively. If A is stable and (A, C) is observable, y will be a stationary stochastic process. We can make use of the Kalman filter to obtain the least mean square estimation of state variable

$$\dot{\hat{x}} = A\hat{x} + B(y - C\hat{x}) \tag{7.149}$$

where B is the Kalman filter gain and

$$B = \Sigma C^T V^{-1} \tag{7.150}$$

Matrix Σ is the solution of

$$A\Sigma + \Sigma A^T + W - BVB^T = 0 \tag{7.151}$$

Define $\varepsilon = y - C\hat{x}$, which is an innovation variable in terms of statistical linear system theory, satisfying

$$E\{\varepsilon) = 0,\ E\{\varepsilon\varepsilon^T\} = V \tag{7.152}$$

This is a property of white noise. Therefore, we can arrive at an equivalent stochastic linear system called innovation representation:

$$\dot{x} = Ax + B\varepsilon \tag{7.153}$$

$$y = Cx + \varepsilon \tag{7.154}$$

The following identification procedure is based on the innovation description.

In OCM, the feedback channel includes time delay and a state predictor. This structure is, however, over-parameterized and hence nonidentifiable. The earlier approach to derive numerical values for model parameters is through heuristic arguments and empirical data [9]. In order to realize identification, we proceed with the canonical form. We now eliminate the time delay assumption in observation. Let the controlled plant be described by

$$\dot{x}_p = A_p x_p + B_p(u + u_w) \tag{7.155}$$

where $x_p \in R^n$ is the state variable vector, and $A_p \in R^{n\times n}$ and $B_p \in R^{n\times 1}$ are already transformed into the canonical form

$$A_p = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ -\alpha_1 & -\alpha_2 & \cdots & -\alpha_n \end{bmatrix},\ B_p = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{7.156}$$

Also, u_w is scalar input disturbance, which is obtained from a dynamic system described by

$$\dot{x}_w = A_w x_w + B_w w \tag{7.157}$$

where $x_w \in R^l$ is the state variable vector in disturbance dynamics, w is the white noise with zero mean and intensity matrix W, and $A_w \in R^{l\times l}$, $B_w \in R^{l\times 1}$ are in canonical form

$$A_w = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ -\beta_1 & -\beta_2 & \cdots & -\beta_l \end{bmatrix},\ B_w = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{7.158}$$

The first component of the state vector is output is

$$u_w = [1 \quad 0 \quad \cdots \quad 0]x_w = C_w x_w \tag{7.159}$$

where $C_w \in R^{1\times n}$ is the output matrix. The composite system with augmented state $x = \begin{bmatrix} x_w^T & x_p^T \end{bmatrix}^T \in R^{(n+l)}$ can be described by

$$\dot{x} = Ax + Bu + Ew \tag{7.160}$$

where

$$A = \begin{bmatrix} A_w & 0_{l\times n} \\ B_p C_w & A_p \end{bmatrix}, \; B = \begin{bmatrix} 0_{l\times 1} \\ B_p \end{bmatrix}, \; E = \begin{bmatrix} B_w \\ 0_{n\times 1} \end{bmatrix} \tag{7.161}$$

The output equation of the HO observation is

$$y = Cx + v_y \tag{7.162}$$

where $C \in R^{1\times(n+l)}$ is the output matrix, and v_y is the observation noise with statistics

$$E\{v_y\} = 0, \; E\left\{v_y(t_1)v_y^T(t_2)\right\} = V_y\delta(t_1 - t_2) \tag{7.163}$$

We assume that the HO seeks to minimize

$$J(u) = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T (x^T Q x + r u^2)\,\mathrm{d}t\right\} \tag{7.164}$$

There is no control rate term in the performance measure. Straightforward use of the standard LQG theory gives the state feedback law

$$u = -kx = -\lfloor k_w \quad k_p \rfloor x = -k_w x_w - k_p x_p \tag{7.165}$$

where $k \in R^{1\times(n+l)}$ is the optimal state feedback gain matrix, and $k_w \in R^{1\times l}$ and $k_p \in R^{1\times n}$ are its partitions corresponding to x_w and x_p. The resultant closed-loop system is

$$\dot{x} = Ax - Bkx + Ew = (A - Bk)x + Ew = A_{cl}x + Ew \tag{7.166}$$

$$y = Cx + v_y \tag{7.167}$$

where A_{cl} is the closed-loop system matrix, a stable matrix from LQG theory [10]. Because the direct state feedback is not available, we construct a Kalman filter to generate the state estimation, which is necessary for state feedback in view of the separation principle. Hence,

$$\dot{\hat{x}} = A_{cl}\hat{x} + K\varepsilon \tag{7.168}$$

$$y = C\hat{x} + \varepsilon \tag{7.169}$$

where $\hat{x}$ is the optimal estimation of x, and K is the Kalman filter gain, satisfying

$$K = \Sigma C^T V_y^{-1} \tag{7.170}$$

where Σ can be obtained by solving the Riccati equation

$$A_{cl}\Sigma + \Sigma A_{cl}^T + EWE^T - \Sigma C^T V_y^{-1} C\Sigma = 0 \tag{7.171}$$

Now we use the state estimation $\hat{x}$ in lieu of x in the state feedback law and include the motor noise term v_{m}; thus,

$$u = -k\hat{x} + v_{\mathrm{m}} \tag{7.172}$$

Here v_{m} is zero mean with intensity V_{m}. Note that Equations 7.167 and 7.171 together form a new dynamic system with ε as the input and u as the output. We again have a new innovation representation given by

$$\dot{\tilde{x}} = A_{cl}\tilde{x} + G\eta \tag{7.173}$$

$$u = -k\tilde{x} + \eta \tag{7.174}$$

where η is the innovation term, and G is the Kalman gain,

$$G = \tilde{\Sigma} k^T V_{\mathrm{m}}^{-1} \tag{7.175}$$

and $\tilde{\Sigma}$ is the solution of the following Riccati equation:

$$A_{cl}\tilde{\Sigma} + \tilde{\Sigma} A_{cl}^T + KV_y K^T - \tilde{\Sigma} k^T V_{\mathrm{m}}^{-1} k\tilde{\Sigma} = 0 \tag{7.176}$$

The statistics of innovation η are

$$E\{\eta\} = 0,\ E\{\eta(t_1)\eta^T(t_2)\} = V_{\mathrm{m}}\delta(t_1 - t_2) \tag{7.177}$$

Now examine the entries in the system matrix A_{cl} to see if it is identifiable.

$$A_{cl} = A - Bk = \begin{bmatrix} A_w & 0_{l\times n} \\ B_p C_w & A_p \end{bmatrix} - \begin{bmatrix} 0_{l\times 1} \\ B_p \end{bmatrix} \begin{bmatrix} k_w & k_p \end{bmatrix} = \begin{bmatrix} A_w & 0_{l\times n} \\ B_p C_w - B_p k_w & A_p - B_p k_p \end{bmatrix} \tag{7.178}$$

where

$$B_pC_w - B_pk_w = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ -k_{w1} & -k_{w2} & \cdots & -k_{wl} \end{bmatrix}$$

and

$$A_p - B_pk_p = \tilde{A}_p = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1 \\ -\tilde{\alpha}_1 & -\tilde{\alpha}_2 & \cdots & -\tilde{\alpha}_n \end{bmatrix}$$

where $\tilde{\alpha}_i = \alpha_i + k_{pi}$, $i = 1,2,\ldots n$. Only the last row of A_{cl} contains unknown parameters, and there are $n + l$ unknown parameters. We now use the nonsingular matrix T to transform $(A_{cl}, -k)$ to invariant canonical form as

$$\tilde{A}_{cl} = TA_{cl}T^{-1}, \tilde{C} = -kT^{-1} \tag{7.179}$$

where

$$\tilde{A}_{cl} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1 \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{n+l} \end{bmatrix}, \tilde{C} = [\ 1 \quad 0 \quad \cdots \quad 0\ ] \tag{7.180}$$

The nonsingular transform T can be derived as

$$T = \begin{bmatrix} -k \\ -kA_{cl} \\ \vdots \\ -kA_{cl}^{n+l-1} \end{bmatrix} = [\ T_1 \quad T_2\ ] \tag{7.181}$$

where

$$T_2 = \begin{bmatrix} -k_p \\ -k_p\tilde{A}_p \\ \vdots \\ -k_p\tilde{A}_p^{n+l-1} \end{bmatrix} \tag{7.182}$$

It can be seen from Equation 7.180 that

$$\tilde{A}_{cl}[\,T_1 \quad T_2\,] = [\,T_1 \quad T_2\,]\begin{bmatrix} A_w & 0_{l\times n} \\ B_pC_w - B_pk_w & \tilde{A}_p \end{bmatrix} \Rightarrow \tilde{A}_{cl}T_2 = T_2\tilde{A}_p \tag{7.183}$$

The last row of the above equation is

$$k_p\left(\tilde{A}_p^{n+l} - \gamma_1 - \gamma_2\tilde{A}_p - \cdots \gamma_{n+l}\tilde{A}_p^{n+l-1}\right) = 0 \tag{7.184}$$

The dimension of $\tilde{A}_p$ is n. Therefore only n of the γ_is are independent in terms of the Cayley-Hamilton theorem [15]. The following are the characteristic polynomials for corresponding matrices:

$$\lambda(A_w) = s^l - \beta_l s^{l-1} - \beta_2 s^{l-2} - \cdots \beta_l, \lambda(\tilde{A}_p) = s^n - \tilde{\alpha}_1 s^{n-1} - \cdots \tilde{\alpha}_n \tag{7.185}$$

$$\lambda(A_{cl}) = s^{l+n} - \gamma_1 s^{l+n-1} - \cdots \gamma_{n+l} \tag{7.186}$$

Because a similar transform does not change the eigenvalues, we have

$$\lambda(\tilde{A}_{cl}) = \lambda(A_w)\lambda(\tilde{A}_p) \tag{7.187}$$

The optimal feedback gain is embedded in the coefficients of the polynomials $\lambda(A_w)$ and $\lambda(\tilde{A}_p)$, but the coefficients in the polynomial of $\lambda(\tilde{A}_{cl})$ is not independent. Therefore, the system, specifically the optimal feedback gain, is not identifiable.

Hitherto we have used OCM, MOCM for HO modeling. As claimed before, they are structure oriented, where the components of state estimate, predictor, and feedback gain integrated together to form the HO model. Interactive work is needed to turn the performance measure and noise intensity to fit the resultant HO model with the measured data. They are not data based. The foregoing discussion is targeted on the identification problem, but its inherited structure from OCM and MOCM prevent it from being identifiable. Next we will develop a new HO model, which is data based and does not have the problem of identifiability.

7.5.9 Data-Based HO Model Identification

The standard solution of the regulator problem for a linear time-invariant system with white noise disturbance corresponds to the LQG theory [10]. When the plant is given and the statistics of the noise are known, its LQG controller can be determined without using empirical data, that is, can be determined off-line. It is optimal in the sense of a user-specified performance measure. In the HO control problem, we regard that it is the competence of the HO that deteriorates the optimality of the LQG controller. We therefore lump the time delay, motor dynamics, and noise into a block called the deteriorating filter, which is cascaded with the LQG controller. The overall system is shown in Figure 7.13. The problem is to identify the deteriorating filter through empirical data. We now illustrate the procedure by an example.

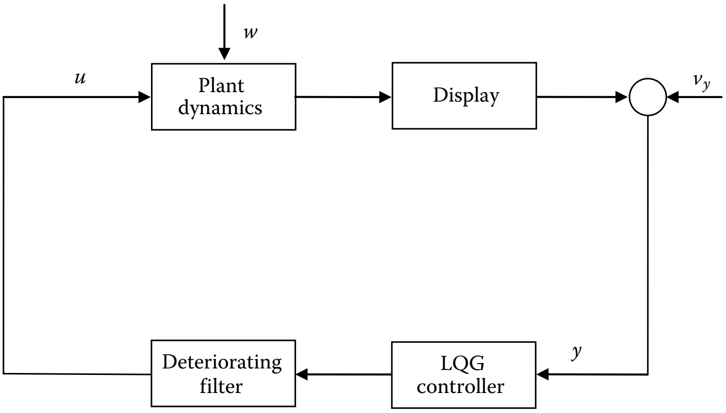

w
u
Plant dynamics
Display
v_y
Deteriorating filter
LQG controller
y

FIGURE 7.13
Data-based HO model.

With the same device as the previous experiment, we now identify an HO model in the setting of Figure 7.14. Where the controlled plant is

$$P(s) = \frac{1}{s} \tag{7.188}$$

The objective of the HO is to use a joystick to keep the output of $P(s)$ near the zero position under the disturbance input, which is the output of

$$F(s) = \frac{1}{s+2} \tag{7.189}$$

driven by a white noise signal $w(t)$ with unit power. HO can only view the error signal on the display. The composite system can be written in the state space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w = Ax + B_u u + B_w w \tag{7.190}$$

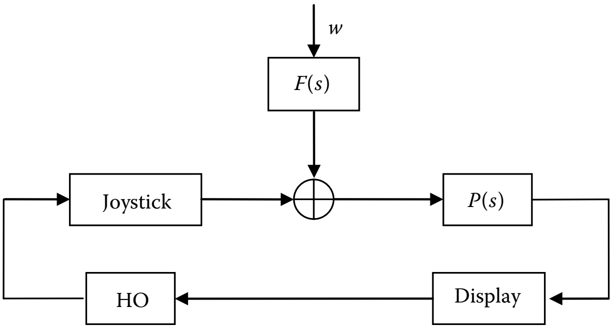

w
F(s)
Joystick
P(s)
HO
Display

FIGURE 7.14
Experiment for the data-based HO model.

$$y = [\ 1 \quad 0\ ] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v = Cx + v \tag{7.191}$$

Suppose that the HO seeks to minimize the performance measure:

$$J(u) = E\left\{ \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(x_1^2 + u^2\right) \mathrm{d}t \right\} \tag{7.192}$$

The control task is in fact the disturbance rejection. The Control System Toolbox in MATLAB provides the function `lqg`, which generates a controller in the state space form, given by

$$\dot{x}_c = A_c x_c + B_c y \tag{7.193}$$

$$u_c = C_c x_c + D_c y \tag{7.194}$$

where

$$A_c = \begin{bmatrix} -1.237 & 0.667 \\ -0.05341 & -2 \end{bmatrix}, B_c = \begin{bmatrix} 0.3268 \\ 0.05341 \end{bmatrix}, C_c = [\ -1 \quad -0.3333\ ], D_c = 0$$

The empirical data taken from one run of the experiment is shown in Figure 7.15. Here the operator controls the plant with a joystick in 100 sec. The result shows that the control, although not perfect, is real.

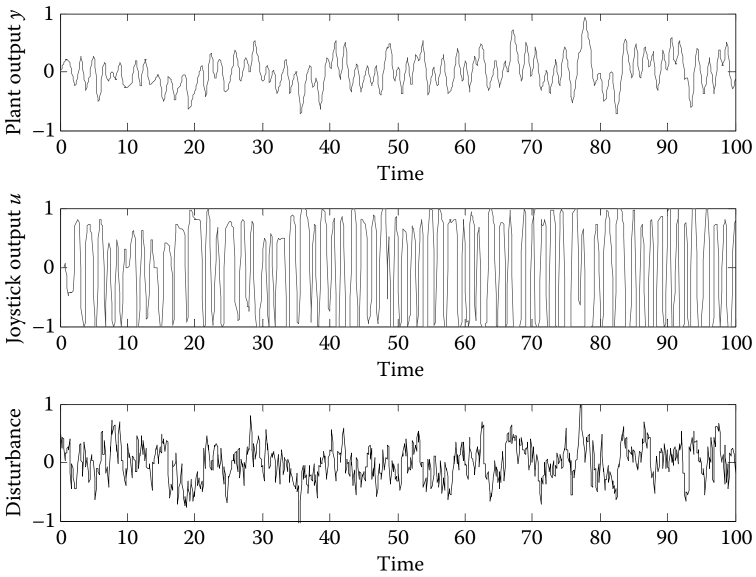


FIGURE 7.15
Disturbance rejection by HO.

The output $y(t)$ is the input to the LQG controller Equations 7.192 and 7.193. By simulation, we obtain the output signal $u_c(t)$, shown in Figure 7.16. The output $u_c(t)$ is the input to the deteriorating filter with the output of the joystick signal. We can now identify this filter in state space form or transfer function form. By using function `ssest` in the System Identification Toolbox of MATLAB and specifying a second-order system, we obtain the deteriorating filter with the following state space description:

$$\dot{x}_\mathrm{d} = \begin{bmatrix} -6.341 & 6.659 \\ -6.383 & 3.013 \end{bmatrix} x_d + \begin{bmatrix} -15.51 \\ -11.21 \end{bmatrix} u_\mathrm{c} \tag{7.195}$$

$$u = [-4.907 \quad -0.7729] x_d \tag{7.196}$$

With $u_c(t)$ as the input, the output of the deteriorating filter is shown in the middle part of Figure 7.16. The lower part gives the difference between the filter output and the actual joystick output. We can deduce that this difference can be modeled further by the remnant noise model.

The identification results indicate the HO model performs far better than the linear model, particularly for less trained HO and complex forcing functions. Usually the random signal employed for an HO model is the sum of several sinusoidal functions with different frequencies and amplitudes, and the HO can perform well. Then, the HO model can be approximated by a linear model. The random signal adopted in this experiment is difficult for the HO to follow. It is why we cannot use the liner model to satisfactorily fit the experimental data.

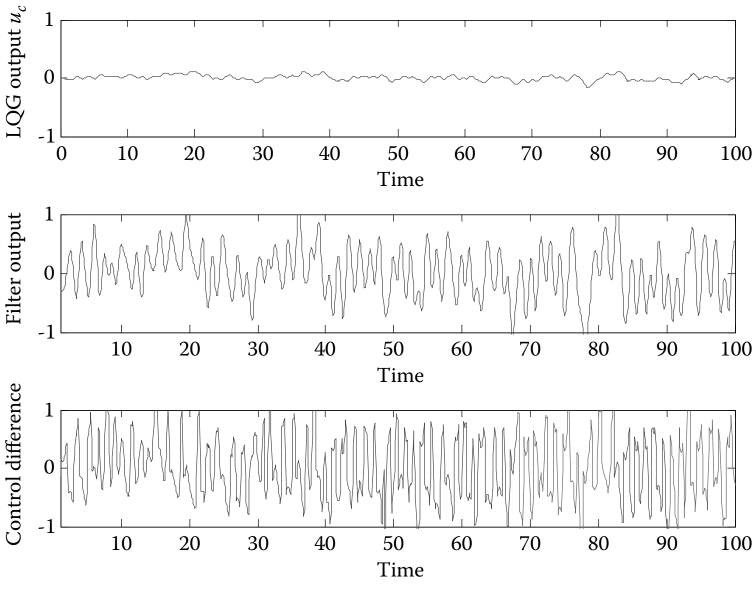


FIGURE 7.16
Signals after LQG controller.

7.6 Conclusions

HO model building is essential for HAM. The long history of pilot or vehicle system research does not mean we can use the related achievements in a straightforward manner here. This chapter investigated one aspect of HAM—system identification for an HO model. If our goal was to find a model to fit empirical test data, classical system identification methodologies would be adequate. But we need to go further. What really concerned us was the mechanism of HO behavior during action. Therefore, an HO model should be highly structured to reflect the optimality and limitations of the HO, the complexity of the control task and the effects of the environment. Quasi-linear and optimal control model were proposed in this context. Parameter estimation under these structures still lack a systematic approach that can lead to a good model.

References

1. Tervo, K., *Human adaptive mechatronics methods for mobile working machines*, PhD thesis, Helsinki University of Technology, Control Engineering, Helsinki, Finland, 2010.
2. Harashima, F., "Human adaptive mechatronics—Interaction and intelligence," *Emerging Technologies Factory Automation, Proc. 10th IEEE Conference on ETFA*, Catania, p. 3, 2005.
3. Sheridan, T. B. and Ferrell, W. R., *Man-Machine Systems: Information, Control and Decision*, The MIT Press, Cambridge, MA, 1974.
4. McRuer, D. T. and Krendel, E. S., "Mathematical models of human pilot behavior," NATO Advisory Group for Aerospace Research and Development, AGARDograph, Brussels, Belgium, p. 188, 1974.
5. Doman, D. B. and Anderson, M. R., "A fixed-order optimal control model of human operator response," *Automatica*, Vol. 36, No. 3, pp. 409–418, 2000.
6. McRuer, D. T., "Human dynamics in man-machine systems," *Automatica*, Vol. 16, pp. 237–253, 1980.
7. Ljung, L., *System Identification: Theory for the User* (2nd Ed.), Prentice Hall, Upper Saddle River, NJ, 1999.
8. Proakis, J. G. and Manolakis, D. G., *Digital Signal Processing* (4th Ed.), Publishing House of Electrical Industry, Beijing, China, 2007.
9. Kleinman, D. L., Baron, S., and Levinson, W. H., "An optimal control model of human response-Part I: Theory and validation," *Automatica*, Vol. 6, No. 3, pp. 357–369, 1970.
10. Anderson, B. D. O. and Moore, J. B., *Optimal Control: Linear Quadratic Methods*, Prentice Hall, Englewood Cliffs, NJ, 1990.
11. Baron, S., Kleinman, D. L., and Levinson W. H., "An optimal control model of human response-Part II: Prediction of human performance in a complex task," *Automatica*, Vol. 6, No. 3, pp. 371–383, 1970.
12. Kleinman, D. L., "Optimal control of linear systems with time delay and observation noise," *IEEE Automatic Control*, Vol. 14, No. 5, pp. 524–527, 1969.
13. Thompson, P. D. and McRuer, D., "Comparison of the human optimal control and crossover models," *Guidance, Navigation and Control Conference*, Minneapolis, MN, pp. 1083–1090, 1988.
14. Davidson, J. B. and Schmidt, D. K., "Modified optimal control pilot model for computer aided design and analysis," Technical Report, NASA, Hampton, VA, 1992.
15. Phatak, A., Weinert, H., Segall, I., and Day, C. N., "Identification of a modified optimal control model for the human operator," *Automatica*, Vol. 12, No. 1, pp. 31–41, 1976.

8

Intelligent Robotic Systems

Muhammad Tahir Khan, Clarence W. de Silva, and Javaid Iqbal

CONTENTS

8.1 Introduction 296
8.2 Biological Immune System 297
  8.2.1 Jerne's Idiotypic Network Theory 299
8.3 Artificial Immune System (AIS) 300
  8.3.1 Network Theory Model 300
8.4 Multi-Robot Cooperation Problem 301
  8.4.1 Fault Tolerance 302
  8.4.2 Decision Conflicts 302
  8.4.3 Interdependencies and Priorities 303
8.5 Multi-Robot Cooperation and Artificial Immune System 305
  8.5.1 Binding Affinity 305
  8.5.2 Robot and Antibody 306
  8.5.3 Multi-Robot Cooperation and Modified Idiotypic Network Model 306
8.6 Genetic Algorithm 308
  8.6.1 Operators of GA 309
  8.6.2 Simple GA 310
8.7 Optimizing Binding Affinity Function Using GA 310
8.8 Results and Discussion 311
8.9 Conclusions 318
References 319

SUMMARY This chapter presents the development of an intelligent multi-robot system. Robots in a multi-robot system *coordinate and cooperate* with each other when they accomplish global tasks. In this chapter, we present a system that is fully autonomous, distributed, and fault-tolerant. The system autonomously decides the appropriate number of suitable and most capable robots in the team to cooperate with each other in carrying out a global task while resolving partial or full failures of the robots during the execution of the task. In our system, the strategies of communication, coordination, and cooperation are based on the principles of human immune systems and the modified model of Jerne's idiotypic network theory. A genetic algorithm is used to optimize the binding affinity function that is needed for selection of the most appropriate robot in a fleet to cooperate with the robot that solicits help. The feasibility of the scheme is demonstrated by implementing the system in a team of simulated heterogeneous mobile robots that cooperatively transport multiple objects to a goal location.

8.1 Introduction

An intelligent multi-robot system (IMRS) involves two or more dynamic robots interacting in a cooperative way for task accomplishment. It is a fully distributed and self-organizing system that obtains a consummate solution for the problem without any external intervention. Robots in an IMRS are somewhat intelligent, possessing local decision-making capabilities. They may cooperate, communicate, and compete with each other. They sense changes in the environment and other robots and take actions based on that information.

While working as a team, a single robot can be more effective than when working alone. For example, it can compensate for the shortcomings of another robot and can perform certain tasks that cannot be performed by a single robot. Cooperatively, a group of robots may be able to carry heavy loads that individual robots might not be able to handle. A multi-robot system is generally more robust and reliable. For instance, if one robot fails, another robot may be able to replace it in order to complete the task. In this manner, the failure of one robot will not cripple the entire system, and the task will be completed, possibly at a lower efficiency. Multi-robot systems are likely to be more efficient because they can work in parallel. In addition, they can achieve diversified tasks by properly teaming up with other robots in the system. Furthermore, tasks needing more than one robot may be attended to rapidly, thereby reducing the task completion time and increasing the overall system efficiency. Multi-robot operation can increase the system flexibility, by taking advantage of the inherent parallelism and redundancy. Furthermore, the complexity of some tasks may require the use of multiple robots as the necessary capabilities or resources may be too expensive to be provided by a single robot. It may be economical to design several less expensive robots with diverse capabilities to work cooperatively rather than designing one complex robot having all the required capabilities. Such a single complex robot is typically less viable from the maintenance and control perspectives.

Developing an IMRS that can execute an allocated task autonomously can be a tremendous challenge. Many of the challenges associated with an IMRS do not occur in a single-robot system; for example, how would the robots settle conflicts? How would the robots interact with each other? How would the robot(s) respond to a failure of an individual robot in a fleet during a mission? The robotics research community has been actively addressing these issues in different domains, for example, multi-agent designs, multi-robot learning, communication and coordination, and cooperation between robots.

An IMRS may be used in a variety of tasks, such as transportation [1], planetary exploration [2], search and rescue [3], hazardous waste cleanup [4], construction [5], and underwater applications [6]. In a nutshell, from the deep ocean to the Earth and from Earth to space, IMRS can be used to handle various challenging tasks. Although multi-robot systems have become an important and highly active research area since the 1980s, the concept of multiple robotic systems was first presented by Walter's robots Elsie and Elmer in the early 1950s. Later, Brooks [7] proposed the subsumption architecture for multi-robot systems as an alternative to the sense-plan-act approach. This was a purely reactive task-based layered architecture in which each layer was composed of a set of behaviors. Matsumoto et al. [8] and Asama et al. [9,10] developed a distributed and autonomous multi-robot system called ACTRESS, focusing on the issues of communication, planning, and task assignment among robots.

In the context of multi-robot systems, an essential feature is cooperation among different team members. Cooperation takes place when the team members perform tasks together in a systematic manner to achieve some global task that either cannot be completed by an individual or whose execution, performance, and efficiency can be improved by using more than one team member. In such a system, an individual does not just perform the task allocated to it but also looks around and observes other team members to determine if there are more important and urgent tasks that need to be carried out. Sometimes an individual may postpone its task and help other team members in order to improve efficiency.

This chapter presents the underlying methodology of using the concepts of a biological immune system in a cooperative multi-robot system. In this methodology, a team of autonomous robots make independent decisions, coordinate, resolve conflicts, and if needed, cooperate with each other to achieve a common goal in an unfamiliar and unpredictable environment with both dynamic and randomly distributed static obstacles. The system autonomously decides the number of robots needed for the task based on its requirements. The feasibility of the scheme is demonstrated by implementing the approach on a team of mobile robots that cooperatively transport multiple objects to a goal location. Subsequently, a genetic algorithm is used to optimize a binding affinity function that determines the most suitable robot within the fleet that can transport or help another robot to transport the object to the goal location. A priority assignment method is also developed for different objects based on ordering dependencies. Decision conflicts among the robots are also discussed in the chapter, and solutions to resolve them are provided.

8.2 Biological Immune System

The biological immune system is a combination of cells, tissues, and organs that work together to protect the human body from the attack of invaders called pathogens. The human body provides an ideal environment for many pathogens, such as bacteria, parasites, and viruses, which cause infections. It is the responsibility of the immune system to keep the pathogens out or eliminate them. The immune system is very complex and amazingly organized. It can recognize and remember millions of different enemies. It contains cells that produce secretions to match up with and destroy nearly all types of invading pathogens.

The biological immune system consists of many cells, including lymphocytes, phagocytes, and their variants. Phagocytes are large in number, and they digest and process pathogens when they enter the body. Phagocytes present the processed part of pathogens, which attracts the attention of the matching lymphocytes. The lymphocytes can be divided into two main types: B lymphocytes and T lymphocytes.

The B lymphocytes or B cells work mainly by producing and releasing substances called antibodies into the blood. Antibodies attack foreign antigens that move in the bloodstream. Each B cell is programmed to produce one specific type of antibody. When a B cell is triggered by a specific type of antigen, it becomes active and gives rise to many large cells called plasma cells, which secrete antibodies.

The T cells mature into helper T cells and cytotoxic T lymphocytes (CTLs), also called killer T cells. The helper T (Th) cells stimulate the immune responses by communicating with other cells. Some of them activate other B cells to secrete antibodies. Killer T cells perform a different function. They directly attack infected cells that have been contaminated by the foreign antigens that have evaded the B cells.

The B cells mature in the bone marrow that produces a Y-shaped protein called an antibody, which is responsible for identifying and neutralizing foreign antigens. As shown in Figure 8.1a, an antibody consists of two identical heavy chains and two identical light

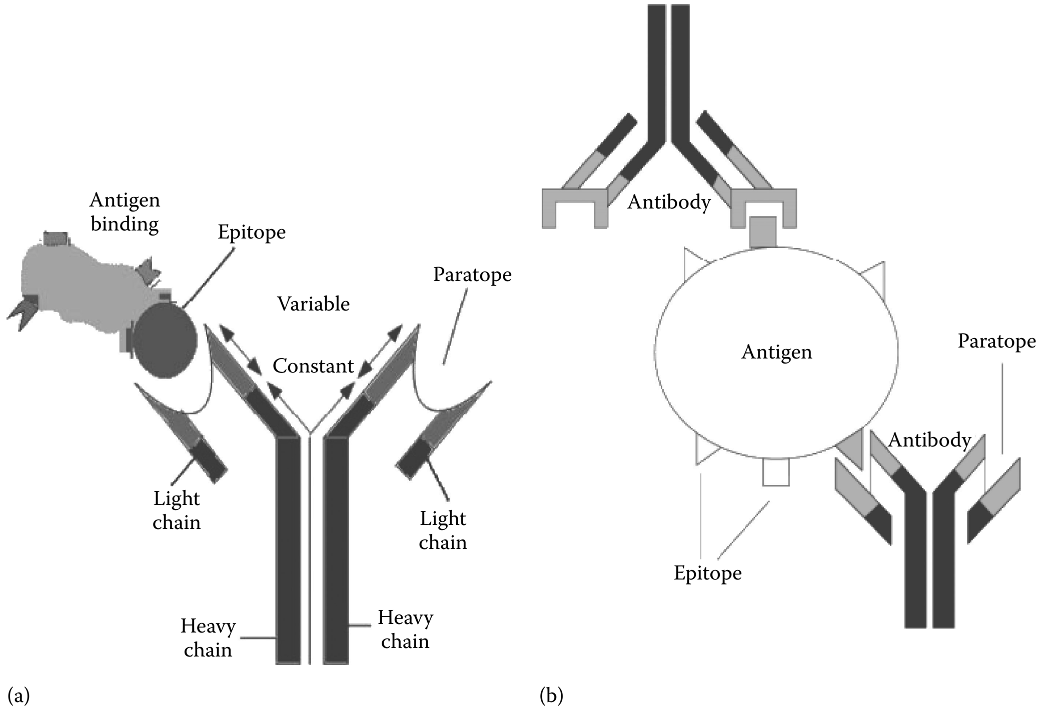


FIGURE 8.1
(a) Antibody structure. (b) Cooperative elimination of antigen by antibodies.

chains joined together to form a Y-shaped molecule. Each chain is composed of variable (V) and constant (C) regions. The two arms of the Y-shaped molecule contain sites that can bind an antigen and are therefore responsible for recognizing foreign objects. Each arm is composed of one variable and one constant domain from each light chain and heavy chain of the antibody. The region formed by the arms of the antibody is called the fragment antigen binding (FAB) region. The upper end of the arms comprises variable domains from the light and heavy chain shapes called the paratope. The paratope is the most important region because it binds to an antigen to produce the immune response. The stem of the Y, known as the fragment crystallizable region (FC), plays an important role in modulating the immune cell activity. This region is composed of two heavy chains, which comprises constant domains. It also determines biological activity, which ensures that the antibody generates an appropriate immune response for a given antigen.

As shown in Figure 8.1a, the binding region on an antigen is called the epitope, which determines the identity of the antigen. Once an antigen enters the human body, the B cells are stimulated, and an antibody whose paratope is complementary to the epitope attaches an antigen to neutralize or eliminate it alone or cooperatively with the help of another antibody.

The antibody receptor recognizes an antigen that has a specific affinity. The binding strength between the paratope and the epitope is proportional to the affinity. A paratope may or may not completely complement an epitope, resulting in a weaker binding affinity. The higher the affinity, the stronger the binding and, hence, the better the immune response to neutralize or eliminate the antigen.

The recognition stimulates proliferation and differentiation of the cells that produce matching clones or antibodies. This process, called clonal expansion, generates a large population of antibody-producing cells that are specific to the antigen. These clones get priority when exposed to similar antigens, which will lead to rapid immune response.

The immune system is highly distributed, decentralized, and cooperative. As shown in Figure 8.1b, if one antibody is unable to eliminate an antigen, more antibodies will cooperate to achieve the objective.

8.2.1 Jerne's Idiotypic Network Theory

In the early 1970s, the eminent Danish immunologist Neil Jerne [11], working in Switzerland, proposed that cells and molecules of the immune system not only recognized foreign substances, but also recognized, responded to, and were regulated by each other. For this reason, the immune system is regarded as a network of interacting cells and antibodies. This viewpoint is known as the idiotypic network theory or simply the immune network theory. The idea is that the cells of an immune system are functionally connected to each other via interaction between the V regions of different antibodies.

Jerne drew a distinction between the parts of the V region of an antibody that is recognized by other V regions of another antibody. He termed the recognizing part of the antibody the "paratope" and the part that is being recognized as an "idiotope." In other words, idiotopes are epitopes in the V regions of antibody molecules.

Figure 8.2 illustrates the immune network theory. It is seen that antibody 2 recognizes the non-self antigen Ag, and the same antibody 2 also recognizes the idiotope Id1 of antibody 1. This means antibody 2 can recognize both the antigen and another antibody. When an antibody's idiotope is recognized by the paratopes of other antibodies, it is suppressed. On the contrary, when an antibody's paratope recognizes the idiotopes of other antibodies or the epitopes of antigens, it is stimulated. The recognition of an antigen by a cell receptor results in network activation and cell proliferation. The network theory entails that B cells are not isolated but rather communicate with each other via dynamic network interaction.

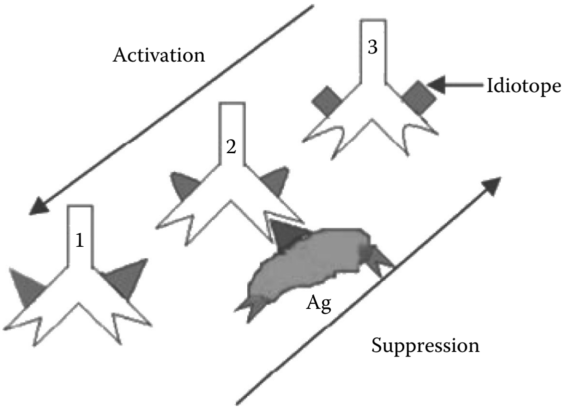


FIGURE 8.2
Jerne's idiotypic network.

8.3 Artificial Immune System (AIS)

Recently, researchers have manipulated the properties of biological systems and espoused the underlying paradigm in various facets of intelligent cooperative multi-robot systems. Humans are known to be enthralled by nature, as evident from Edmund Wilson's sociobiology or cooperative behavior of ants and social insects [12], neural networks based on biological nervous systems, genetic algorithms, programming based on biological evolution [13,14], and fuzzy logic inspired by human knowledge and reasoning [15]. Researchers have suggested numerous biology-inspired solutions for multi-robot cooperation.

Based on the notion of the human immune system, a new computational technique called the artificial immune system (AIS) is rapidly emerging. AIS is a new computational intelligence paradigm, which aims at solving intricate engineering and computational problems. It attempts to bridge the gap between immunology and engineering. Several mathematical and computational models are taken from immunology and abstracted into algorithm design, to investigate and realize solutions for real world science and engineering problems.

AIS emerged in the late 1980s with the earlier theoretical immunology work of Farmer, Packard, and Perelson [16] and Bersini and Varela [17]. In the 1990s, the interest on AIS from the computing community began to rise. Forrest et al. [18] studied self and non-self in an immune system, and Dasgupta conducted extensive studies on negative selection algorithms [19,20]. Dasgupta edited the first book on artificial immune systems in 1999 [21]. Currently, many new ideas, algorithms, and models along AIS lines, inspired by the natural immune system, are being explored.

Unlike established paradigms, such as genetic and evolutionary computation, artificial neural networks, and various forms of artificial life, AIS is still in its infancy. There is no established structure for AIS. Researchers have developed different immune-inspired algorithms and are bringing improvements in them. Widely developed and used AIS algorithms include negative selection, clonal selection, and idiotypic network.

8.3.1 Network Theory Model

The work presented in this chapter is based on the network theory model. In 1986, Farmer et al. [16] modeled the idiotypic immune network in computer simulation. They greatly simplified the immune system in order to focus on the elements that contain the core of the idiotypic network. The network theory was modeled as a differential equation simulating the rate of change of antibody concentration with respect to stimulation, suppression, and natural death rate. The authors assumed in the model that there were N antibody types $\{x_1, x_2, \ldots, x_N\}$ and n antigens $\{y_1, \ldots, y_n\}$. The assumptions have led to the following differential equation:

$$\dot{x}_i = c\left[\sum_{j=1}^{n} m_{ji}x_i y_j - k_1 \sum_{j=1}^{N} m_{ij}x_i x_j + \sum_{j=1}^{N} m_{ji}x_i x_j\right] - k_2 x_i \tag{8.1}$$

An antibody is represented as a pair of binary strings (p, e) with p denoting the paratope string and e the epitope string. The epitope of an antibody is quintessentially an idiotope. The degree of match between the binary strings of the paratope and epitope imitates the binding affinity between a real paratope and epitope and uses the XOR operator to test the degree of matching in the bits of the string. It is not important that all the bits of p and e must match. However, a threshold value s has to be defined, below which they will not react. For example, if s is set to six and there are six matches for a given alignment, then the score will be one; for seven matches, the score will be two, and so on. The bits of p and e can match in any alignment. The strength of all possible reactions between paratopes and epitopes is represented by $G = 1 + \delta$, where δ is the number of matching bits in excess of the threshold value. The degree of reaction in all possible alignments between antibody i and antibody j is represented as $m_{ij} = \sum G$.

The first term within the braces in Equation 8.1 represents the stimulation of antibody i in response to all antigens. The second term represents the suppression of antibody of type i when its epitopes (idiotopes) are recognized by other antibodies. The third term corresponds to the stimulation of antibody i in response to all other antibodies. Also, m_{ij} and m_{ji} denote the matching function that shows which reaction has happened, and its strength c is a rate constant, and k_1 denotes the likely inequalities between stimulation and suppression. Specifically, when $k_1 = 1$, the stimulation and suppression forces are equal. The final term outside the brackets in Equation 8.1 represents the propensity of the cells to die in the absence of any interaction at a rate determined by k_2.

8.4 Multi-Robot Cooperation Problem

In this chapter, an intelligent multi-robot system is developed that is validated through cooperative transportation of multiple objects to a goal location. In the beginning, robots explore the environment by searching for the task. Once a robot finds the task, it attempts to accomplish it by itself, provided that it has the required capabilities. Otherwise, it informs other robots about the presence of the task and starts searching the environment for another task. However, if a robot has the capabilities but is still unable to transport the object alone, it broadcasts a help signal to seek assistance from other available robots. The robot that seeks help is termed the instigating robot. The help signal contains some information about the task and its location. All the robots in the sensory range who are in the explore state will accept the signal. Based on the information about the task, all robots will examine their list of capabilities to determine their suitability for the task. The robots that do not have the required capabilities will change their state to explore and will resume the search for another task. As there may be more than one capable robot that can offer help, the most appropriate robot among them should be chosen. A binding affinity function is used to resolve this conflict as discussed in Section 8.5.1. The robot that goes for help is termed the facilitating robot. The facilitating robot approaches the object to transport it cooperatively with the instigating robot. Once the facilitating robot reaches the object, the two robots (the instigating and facilitating robots) send a synchronizing signal to each other and initiate cooperative transportation of the object. During cooperation, the robots communicate with each other only if required. For example, they will communicate when an object needs to be rotated or if an obstacle is to be avoided.

The failure possibilities of robots during the selection of the most appropriate robot and during the transportation are also addressed. Although the malfunctioning of a single robot may not cripple the entire multi-robot system, it can adversely affect the ability and efficiency of the robotic team. To minimize this effect, the present approach provides a fault-tolerant solution to autonomous multi-robot systems.

8.4.1 Fault Tolerance

Fault tolerance is the ability of the IMRS to detect and respond to (or accommodate or resolve) the failures in individual robots or in communication that may occur at any time during the mission. Fault-tolerant design enables a system to continue its intended operation, possibly with degraded efficiency, rather than failing completely, when some or all parts of the system fail.

Dias et al. [22] studied communication failure, partial robot malfunctioning, and complete robot failure (death) during multi-robot coordination. Parker [23,24] developed a behavior-based, fault-tolerant software architecture inculcating impatience and acquiescence behaviors to handle the failures in robots. Gerkey and Mataric [25,26] developed an auction-based task allocation system, which was robust to robot failures.

Two broad classes of possible malfunctions in multi-robot systems are partial failure and full failure. When partial failure occurs in a robot, it loses the ability to use some of its resources. Partial failures are further classified into two types. In the first type, the robot is able to detect and communicate its failure to the team members. In the second type, the malfunctioning robot is incapable of detecting and communicating its failure to the teammates. The first type of partial failure is reasonably easy to handle; once it communicates its failure, the faulty robot is replaced, and the task is reallocated to a healthy robot in the team. Clearly, it is more difficult for the system to detect the second type of partial failure in a robot. This type of failure may deceive the other members of the team. For example, if a communication link of the malfunctioning robot is operational, it may communicate or coordinate with other robots incorrectly, which may lead to the incorrect belief that the robot has not failed.

In case of full failure (robot death), the robot ceases to work completely; that is, none of its subsystems function. Detecting a full failure is rather difficult because the dead robot can neither detect its own failure nor is it able to reallocate its task.

Robot malfunction can happen at any stage of multi-robot cooperation. In this chapter, the issue of fault tolerance is addressed in some rigor, encompassing both full and partial failures. In particular, the failure of the instigating or the facilitating robot during the selection of the most appropriate robot and failure during execution of the cooperative task are investigated.

8.4.2 Decision Conflicts

In the real world, people constantly make decisions that affect their lives: whether to take public transit or drive, drink coffee or tea, whether to take one job or another. In order to enrich the multi-robot cooperative environment, decision conflicts are incorporated into the robots to mimic decisions that a human would have to make if humans were present in place of the robots. Three main decisions are identified: the in-process decision, the inbound decision, and the aiding decision.

An in-process decision is made when a robot is in the process of transporting an object and receives a message calling for help to transport another object. This call may be the result of either the robot that is calling out for help finding an object that it is not capable of moving or the robot is having difficulty in transporting the object. Either way, the robot that is transporting an object must make a decision as to whether it should abandon its current task and comply with the help request or continue with the current object.

An in-bound decision is made when a robot is in the process of transporting an object and detects another object along its path to the goal. The robot then needs to make a decision whether it should abandon its current object and transport the detected one or continue to transport the current object. The main factor in this decision is priority. In this case the priorities of the two objects should heavily affect whether the robot chooses the current object over the detected one.

The last decision conflict is not about making a choice between the priorities of different objects but rather concerns the capabilities of the transporting robot. The aiding decision is made when a robot is transporting an object with difficulty. Difficulty is defined as transporting an object below a certain speed threshold. When a robot detects that its speed is below the threshold, it sends a message to the other robots in the environment indicating that it requires help for the transportation of its object. Each robot in the communication range then decides whether it is able to help and sends its decision back to the requesting robot. Whether the requesting robot receives help or not is entirely dependent upon the situation.

These decision conflicts may arise separately or in any combination during the mission. Binding affinity (Section 8.5.1) plays an important role in resolving the conflicts.

There is an intricate relationship between the number of robots that are capable of transporting a particular type of object and the number of objects in the environment that require the combined capabilities of robots. That is, if the requesting robot requires capability A for transporting an object, but the surrounding robots have capabilities A and also B, C, and D, then these robots may be required at some stage to transport other objects in the environment that require capabilities A, B, C, and D. A situation can also be imagined in which an aiding robot has come to help after abandoning its previous task. It is to be expected that the object to be transported by both robots has a higher priority than that of the aiding robot's original part—in fact, adequately high to overcome the distance and other parameters as given in Equation 8.3.

8.4.3 Interdependencies and Priorities

We have developed a priority-based method to enrich the decision making of the robots as they complete their tasks. Now an example is given to illustrate the significance of this technique. Suppose a robot at a goal location assembles a mechanical structure from parts located in its unknown environment. Of course, the parts must be retrieved in the order of assembly; however, as the environment is dynamic and incompletely known, the location of the part may be also unknown. As shown in Figure 8.3, the relationship between the parts can be modeled as a graph, on which each node represents a part (e.g., nuts, bolts, shaft, gear) and an arrow represents two parts that connect to each other (e.g., a gear connects to a shaft).

In the beginning, each node starts with the same priority. Let P be the priority of some node in the graph in the beginning of the algorithm, and let N be the number of nodes in the system. The total value of priority within the system is the sum of all the priorities of the graph, which remains constant at the value Np.

Figure 8.3 shows how priorities can be distributed among the tasks after the previous tasks are completed. The nodes A and B represent different parts that can be transported by robots with different capabilities. The value given at a task node represents the task's priority. The priority of the overall system stays constant throughout the simulation. Every

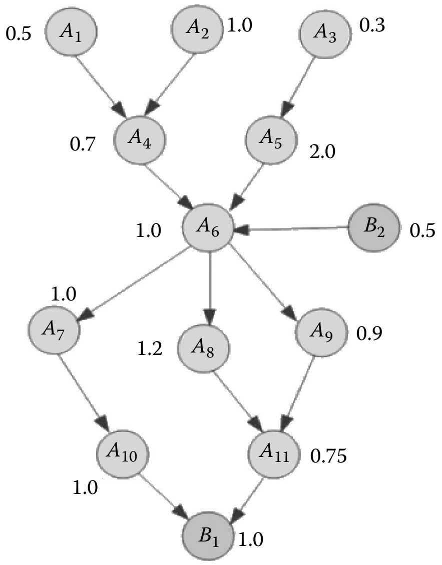


FIGURE 8.3
An example of task distribution.

time a task node P is completed, it is removed from the task list, and the priorities of all the neighbor nodes of P are updated. The new priority of each neighbor is equal to the sum of the neighbor's priorities (Np) and the priority of P (p_{p}) divided by the total number of P's neighbor nodes (N):

$$p_{\mathrm{P}} = \sum_{k=1}^{N} p_k + \frac{p_{\mathrm{P}}}{N} \qquad (8.2)$$

Leaf nodes of the graph act as priority sinks. Priority may flow into a leaf node but cannot flow back into the system. This does not appear to be an issue for the algorithm because once those leaf node tasks are completed, the task is not compared to other tasks to determine the priority, and thus the relative priority ordering of tasks that are yet to be completed is still preserved. The algorithm behaves well no matter where in the graph the first task is completed. In fact, the algorithm behaves well even if the nodes are randomly completed within the graph. As priorities flow from completed tasks to uncompleted tasks, the priorities of uncompleted tasks rise in proportion to the number of completed neighbors. This increase in priority based on the completed neighbors increases the chances that if a node with high enough priority is found, low priority parts will be temporarily abandoned in favor of a high-priority part.

To apply the work flow analogy described above, suppose that a gear and a shaft have been successfully transported, and a bolt is required to connect the components together. In this situation, the priority of the bolt increases after the gear and the shaft have been transported to the assembly location. If the robot detects a bolt but is unable to transport it due to its capability limitations, it will broadcast the message detailing the location of the bolt and its priority to the teammates. If there is a robot nearby that is transporting another part of lower priority, the object may be abandoned in favor of the bolt.

8.5 Multi-Robot Cooperation and Artificial Immune System

The artificial immune system is increasingly finding applications in different research areas of multi-robot systems, such as controls and coordination cooperation. Gao and Wei [27] developed a probabilistic immune agent model-based algorithm for autonomous multi-robot cooperation. Lau and Wong [28] proposed a multi-agent control framework based on an artificial immune system. Simulated robots were used to transport goods to a goal location. Gao and Luo [29] proposed an artificial immune network model for a cooperative multi-robot system.

The immune system is a highly cooperative, distributed, and decentralized system. If one antibody is unable to eliminate an antigen, it cooperates with other antibodies or T cells to achieve the goal. The cooperative multi-robot task execution in the present work is similar to the task of antigen elimination in a human immune system. Specifically, a robot is analogous to an antibody, the robot's capabilities are equivalent to the antibody paratope, a task is analogous to an antigen, and the properties of the task are analogous to the antigen epitope. The environment in which the robots work contains both static and dynamic obstacles, which can be regarded as self antigens. As in the biological immune system, the self-antigens are handled differently from the non-self antigens, and the self antigens (obstacles) too are avoided. However, as a multi-robot environment is dynamic and unknown in general, for simplicity, all elements in the environment other than the tasks are considered to be self antigens. Task completion by the robots is analogous to the antigen elimination by the antibodies. These analogies will be used in the succeeding paragraphs.

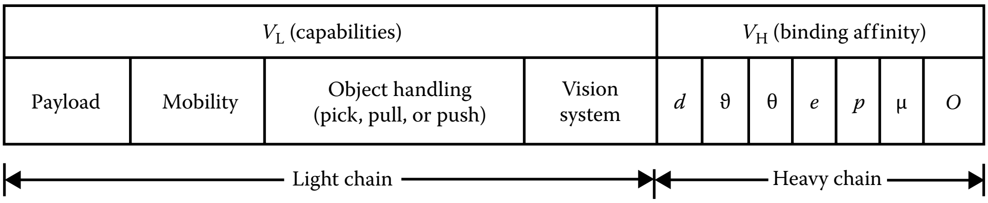
8.5.1 Binding Affinity

Affinity represents the strength with which an antibody paratope binds to an antigen epitope or idiotope. The degree of binding is directly proportional to the elimination level of the antigen. Hence, the binding affinity is used to determine the capable and most appropriate antibody (i.e., the facilitating robot in the present application) among the stimulated antibodies to cooperate with the instigating antibody to eliminate the antigen.

Binding affinity is a function of seven factors: the distance between the antigen and the stimulated antibody, the antibody velocity, the orientation of the antibody with respect to the antigen, the effort required to eliminate the antigen, the antigen priority, the antibody strength, and the obstacles between the antibody and the antigen. The fitness of an antibody with respect to an antigen is the weighted sum of these seven factors as given by

$$\beta^{-1} = w_1(d) + w_2(\vartheta) + w_3(\theta) + w_4(e) + w_5(p) + w_6(\mu) + w_7(O) \tag{8.3}$$

Here w_i are the weights assigned to differed variables, d represents the Euclidean distance between the antibody and the antigen, ϑ is the velocity at which the antibody travels to reach the antigen, θ represents the orientation of the antibody with respect to the antigen, e is the effort required to eliminate (transport) the antigen, p represents the priority of the antigen that needs to be eliminated first, μ is antibody strength, and O represents the obstacles in the path between the antibody and the antigen within the sensory range. *Note*: Not necessarily the antibody that is closest to the antigen will reach it first. Also, an antibody with more capabilities will cost more to operate. For determining the optimal weights, a genetic algorithm is used. This will provide a more methodical solution, as explained in Sections 8.6 and 8.7.



V_L (capabilities)				V_H (binding affinity)						
Payload	Mobility	Object handling (pick, pull, or push)	Vision system	d	ϑ	θ	e	p	μ	O

FIGURE 8.4
Antigenic determinant variable regions of antibody.

8.5.2 Robot and Antibody

The antibody in an immune system has a distinct structure, which resembles the letter Y, containing light and heavy chains. The details are given in Section 8.2. Analogous to an antibody, a robot comprises a pair of variable regions V_H and V_L of light and heavy chains. As indicated in Figure 8.4, the variable region V_L contains the capabilities of the antibody, for example, sensors such as the vision system; hardware capabilities such as arm, bumpers, and gripper; and operational capabilities such as payload, push, pick, and mobility. The variable region V_H represents the binding affinity of the antibody (robot) to the antigen at a particular instance.

8.5.3 Multi-Robot Cooperation and Modified Idiotypic Network Model

Of late, researchers have exploited the properties of the idiotypic network in the field of robotics. A solution for mobile robot navigation based on idiotypic network theory has been proposed in [30]. Luh and Liu [31] have proposed a reactive immune network for mobile robot navigation based on the principle of an immune network. Ichikawa et al. [32] applied the idea of network theory to the gait coordination problem of a hexapod robot. Wang et al. [33] proposed an action selection algorithm for soccer robots by using the concept of the immune network. Li et al. [34] developed a cooperation approach among autonomous robots based on immune response and network theory. Sathyanath and Sahin [35,36] developed an AIS-based multi-agent model named AISIMAM and applied it in mine detection. The communication among the antibodies was based on the immune network theory.

In the present chapter, an antibody deals with one antigen (task) at a time. If the antibody cannot tackle the antigen alone, it will seek help and will coordinate with the facilitating antibody. In applying the Farmer's model (Equation 8.1) to multi-robot cooperation, it has been modified as given by

$$\dot{x} = \left\{ \alpha_{ji} x_i y_j + \sum_{i=1}^{n} \alpha_{ji} x_i x_j - \sum_{j=1}^{N} \alpha_{ij} x_i x_j \right\} - k x_i \qquad (8.4)$$

The values of c in Equation 8.1 are equal to one in the present work.

Figure 8.5 presents an example to explain the idiotypic network-based cooperation. Once the antibody x locates an antigen y (object to be transported), it attempts to eliminate it alone. If unable to do so, it will seek help from other antibodies and will cooperate and coordinate with them to eliminate the antigen (i.e., to transport the object).

In the first step, the antibody matches its paratope (robot capabilities) with the antigen's epitope (partially known properties of the task) as indicated in Figure 8.6. For example, the antigen weight is compared with the payload capability of the antibody, and the shape

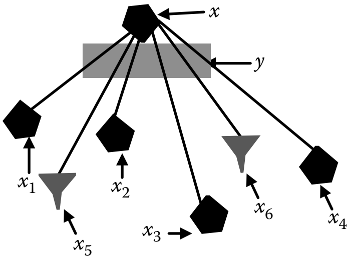


FIGURE 8.5
Antibody–antigen and antibody–antibody stimulation and suppression.

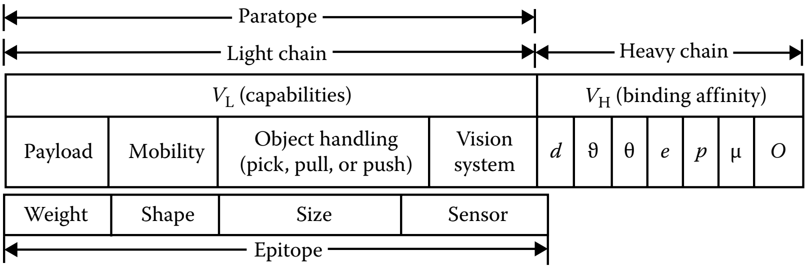


FIGURE 8.6
Matching of antigen epitope and antibody paratope.

and size of the antigen determines the required object handling and motion capabilities of the antibody. The matching is done in full alignment, and the threshold value for the match is equal to the number of partially known properties in the epitope list. If a paratope matches an epitope, the antibody is stimulated, and it will start eliminating (transporting) the antigen. This encounter is indicated in the first term of Equation 8.4.

The first term in Equation 8.4 represents the stimulation of antibody x_i in response to the lone antigen y_j, and α_{ji} is the matching function between the antibody and the antigen. If the antibody x is unable to eliminate the antigen alone, it seeks help from other antibodies. The help signal of antibody x contains its idiotope. An idiotope contains partially known information about the antigen. The help request is received by all antibodies that are in the explore state and within communication range. As shown in Figure 8.5, the robots x_1–x_6 receive the help request. All antibodies compare their paratope with the idiotope of antibody x as indicated in Figure 8.7. The comparison is done in all alignments, and the threshold is equal to the number of antigen epitope properties received in the idiotopes from antibody x. The antibodies x_5 and x_6, which do not have the required capabilities, disregard the help signal. The paratopes of robots x_1–x_4 match the idiotope of x and are stimulated. The stimulation is given by the second term of Equation 8.4.

The second term in Equation 8.4 represents the stimulation of the antibodies x_i, namely x_1–x_4, in response to the antibody x_j, namely x. Here $i = 1 - N$ represents the stimulated robots, and α_{ji} is the matching function, which represents the degree of recognition for stimulation. Unlike in an immune system, here the antibodies communicate with each other without coming into physical contact. In the present case, an antibody is stimulated in response to one individual antibody unlike in the idiotypic network theory in which one antibody can be stimulated in response to many antibodies. Based on the second term of Equation 8.4 and as shown in Figure 8.5, four antibodies are stimulated to help the

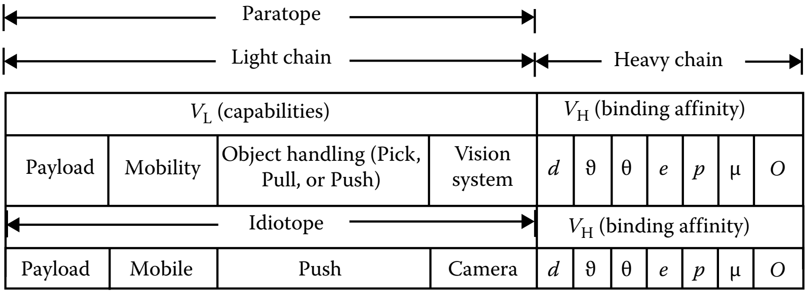


FIGURE 8.7
Antibody–antibody paratope and idiotope matching.

antibody x. One stimulated antibody, the most appropriate one among the four, must go to help, and the rest should go into suppression state and start searching for other tasks. To be selected for cooperation with the instigating antibody, each stimulated antibody calculates its binding affinity (see Section 8.5.1) based on the values of the variable region V_H of the heavy chain. Every stimulated antibody broadcasts its binding affinity to the other stimulated antibodies and compares its own binding affinity with the received one.

The third term in Equation 8.4 represents the suppression of antibody x_i in response to all other antibodies x_j. Here $J = 1 - N$ represents the number of antibodies, and α_{ij} is the matching function representing the degree of recognition for suppression. An antibody x_i receives the value of the binding affinity function from the other stimulated antibodies x_j. There are three such antibodies in the example shown in Figure 8.5. After comparison, the antibodies that have a lower binding affinity than the received one go into suppression. The suppressed antibodies change their status to explore, and they start searching for a new antigen. The antibody having the highest binding affinity, termed the facilitating antibody, is stimulated to help the instigating antibody x, which will cooperatively eliminate the antigen (i.e., transport the object).

This process is extended in a similar manner if a third antibody is required, in case the two antibodies are unable to eliminate the antigen. The instigating or facilitating antibodies may fail fully or partially at any stage of coordination and cooperation.

In the event of malfunction of the instigating or facilitating antibody, the other antibody must take corrective measures to replace the failed antibody with a healthy one. To do this, the malfunctioned antibody must first be declared failed by itself or the collaborator antibody. Next, the selection process of another healthy antibody should be started as explained in this section. The antibody is declared failed based on the last term of Equation 8.4. Here k is the stimulus rate at which the malfunctioning antibody x_i is declared failed. Note that k varies depending on the type of failure, full or partial.

8.6 Genetic Algorithm

Computer algorithms that model survival of the fittest are very striking because they are simple to program and do not involve complex mathematical formulations. These algorithms have been applied to a wide variety of optimization problems, leading to some exquisite results.

During the 1950s and the 1960s, several computer scientists studied evolutionary algorithms in order to develop an optimization tool for engineering problems. Genetic algorithms (GAs) were developed and formulated by John Holland in the 1960s. Later in the 1960s and 1970s, GAs were further improved by Holland, his students, and colleagues at the University of Michigan.

A GA is an intellection of biological evolution. It is a technique of moving from one population of "chromosomes" to a new one through "natural selection," together with genetics-inspired operations of crossover and mutation. The selection and recombination operator generates a new population of chromosomes with the hope that the new population will be better than the old one. The new population (offspring) is selected according to fitness. Those chromosomes that represent a better solution to the target problem are given more chances to reproduce as compared to the ones with poorer solutions.

8.6.1 Operators of GA

The simplest form of GA involves three types of operators: selection, crossover (single point), and mutation.

Selection: It selects chromosomes in the population for reproduction. The fitter the chromosome, the greater the likelihood of its selection for reproduction. The chromosome should in some way encompass information about the solution that it represents. Binary string is commonly used for encoding. However, depending upon the problem, there are many other ways of encoding, for example, using integers and real numbers. The binary chromosome may look like what is shown in Figure 8.8.

Crossover: This operator randomly chooses a crossover site along the bit string. The bits of two strings are exchanged up to this point as shown in Figure 8.9. The crossover operator mimics biological recombination between two single chromosome organisms.

Mutation: The mutation operator is applied after recombination. This operator randomly flips some of the bits in a chromosome at some probability. For example, the string 10110010 may be mutated in its third position as 10100010. The purpose of this operator is to maintain diversity within the population and impede premature convergence to a local optimum.

Chromosome 1	10110010
Chromosome 2	01011100

FIGURE 8.8
Binary chromosomes.

Chromosome 1	1011 | 0010
Chromosome 2	0101 | 1100
Offspring 1	1011 | 1100
Offspring 2	0101 | 00100

FIGURE 8.9
Chromosome crossover.

8.6.2 Simple GA

1. **[Start]** Randomly generate random population of *n* chromosomes
2. **[Fitness]** Determine and evaluate the fitness $f(x)$ of each chromosome x in the population
3. **[New population]** Create a new population by repeating the following steps until the new population is complete:
    1. **[Selection]** Select two parents from the population of chromosomes according to their fitness
    2. **[Crossover]** Perform crossover on the parents with a crossover probability to form a new offspring (children)
    3. **[Mutation]** Perform mutation of the offspring at a mutation probability
    4. **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use newly generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in the current population, else
6. **[Loop]** Go to step **2**

8.7 Optimizing Binding Affinity Function Using GA

GAs are actively used in MRS research. Cai and Peng [37] suggested a novel cooperative coevolutionary adaptive genetic algorithm for path planning of a cooperative MRS. Kubota et al. [38] proposed a perception-based genetic algorithm for mobile robots to acquire collision avoidance behaviors. Yang and Luo [39] presented a GA-based algorithm for coalition structure formation in MRS.

To establish the optimized weights, a GA can provide a less intuitive and more methodical approach. During the initialization of the GA, a random weight between 1 and –1 is generated for each weight of Equation 8.3, that is, w_1, w_2, w_3, w_4, w_5, w_6, and w_7. The simulation is then run with a randomly generated set of weights. The set of weights having the best score is noted. After the initialization, the GA is run for a number of generations in which inheritance, selection, crossover, and mutation of the weights occur. The GA terminates if there is only a 1% difference in score of three successive generations. Generally, after each generation, a set of five candidate parents is selected from the pool of weight sets produced by running the simulation. A linear probability distribution function is used to select the best parents for each child to inherit the parents' genes (i.e., the weights). After selecting two of the five candidate parents, the crossover operation is performed. For each parameter or weight, the child has a 50% chance of inheriting the weight from one of the two parents. Finally, with the combined weights, there is a 10% probability of mutation in one of the parameters of the weight set produced from the two parents.

Fifty different scenarios were created to optimize the weight set of binding affinity function with the GA. Each configuration file of a scenario was constructed by specifying the antibody and the antigen to be transported to the goal. A score was assigned to each scenario after the antibody successfully eliminated (transported) all the antigens (parts). The score is dependent on the number of time steps the antibody takes to complete the task.

TABLE 8.1

Optimized Set of Weights

Variable	Weight Value
Distance	0.39
Velocity	−0.61
Orientation	0.98
Effort	−0.69
Priority	−0.14
Strength	0.19
Obstacle	−0.48

The antibodies were given a limit of 5000 time steps to bring all antigens (parts) to the goal, and if the limit was exceeded, the scenario was penalized by 50,000 time steps. The total score S assigned to each trial is the sum of the time steps to complete all 50 scenarios, and t_i is the total number of time steps taken to complete the i_{th} scenario as $S = \sum_{i=1}^{50} t_i$. After running the simulation, the optimized weight set given in Table 8.1 was generated.

8.8 Results and Discussion

The feasibility of the developed scheme is tested on a team of mobile robots transporting multiple objects to a goal location in an unknown, unpredictable, and dynamic environment. To properly illustrate the present work, the following assumptions are made on the developed methodology:

1. The robots are heterogeneous, possessing different capabilities and sensors.
2. A robot may fail partially or fully at any stage and at any time. It may or may not be able to communicate its failure to the team members.
3. A robot has a local view and does not have any knowledge of the overall system and the environment.
4. The system is completely distributed. There is no leader or controlling robot available to monitor the progress or the state of other team members, the mission, or the environment. No central knowledge is available.

The multi-robot test environment with control software for the transportation of multiple objects is developed in the present work. It consists of simulated heterogeneous mobile robots and objects with different properties to be transported to a goal location. The environment also contains randomly distributed static obstacles. Each robot is a dynamic obstacle to other robots.

The present study addresses two aspects: (a) The selection process of the most suitable antibody for cooperation, if needed, and (b) cooperative elimination (transportation) of an antigen (object).

During the process of selecting an appropriate antibody to help in cooperative task execution, failure of an antibody may occur. In case of failure, if there is no response from an antibody or if an antibody does not behave as expected, the collaborating antibody, which provides stimulation, will come to know of the problem. The stimulation increases with time, and once the stimulus level *k* (Equation 8.4) reaches the threshold value, the collaborating antibody will declare the malfunctioning antibody as failed. The threshold depends on the type of failure, whether partial or full. Subsequently, the collaborating antibody will rebroadcast the help signal.

Figures 8.10 through 8.15 present the results of antibody failures, their effect on the system performance, and the robustness of the approach. These results are based on the scenario in which more than one antibody is required to cooperatively eliminate an antigen. Both partial and full failures are studied here. The partial failure occurs when an antibody is capable of detecting its breakdown and can communicate that to the teammates. This is named here "Partial failure 1." The partial failure in which the antibody for some reason is unable to detect and communicate its failure to other antibodies is termed "Partial failure 2." In Partial failure 2, it is assumed that the communication channel of the failed antibody is operational.

Figures 8.10 and 8.11 present the results when the facilitating antibody fails before it reaches the antigen that is to be cooperatively eliminated. Figure 8.10 shows the time (steps) incurred in different types of failures. The time (steps) taken in Partial failure 1 is less than that for other types of failures because the malfunctioned antibody is able to communicate its failure to teammates. However, in Partial failure 2 and in full failure, the time (steps) taken is greater because the instigating antibody has to wait longer until the stimulation level *k* reaches the threshold before declaring the failure of the facilitating antibody.

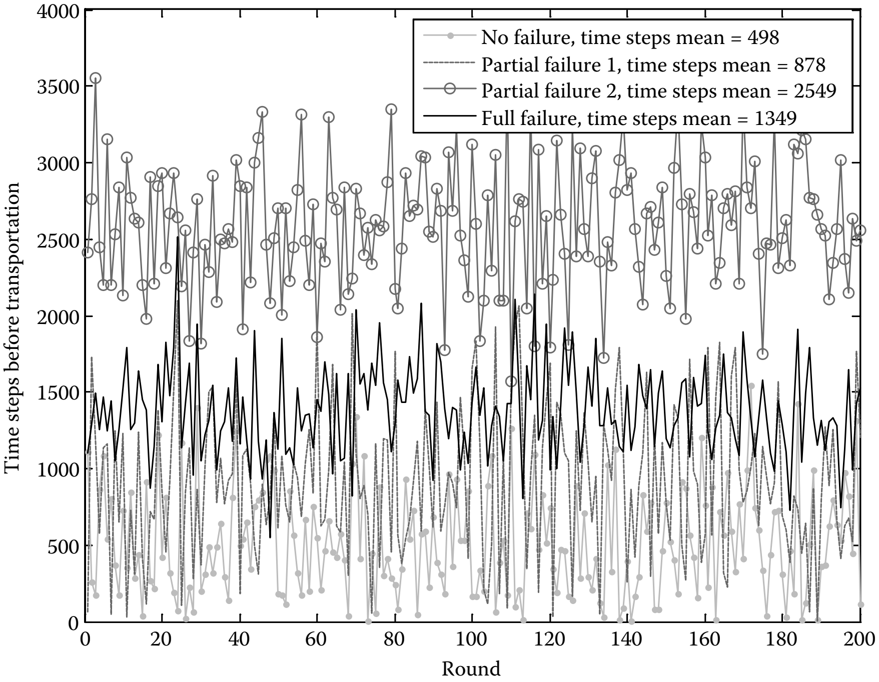


FIGURE 8.10
Effect on time (step) from partial or full failure of facilitating antibody.

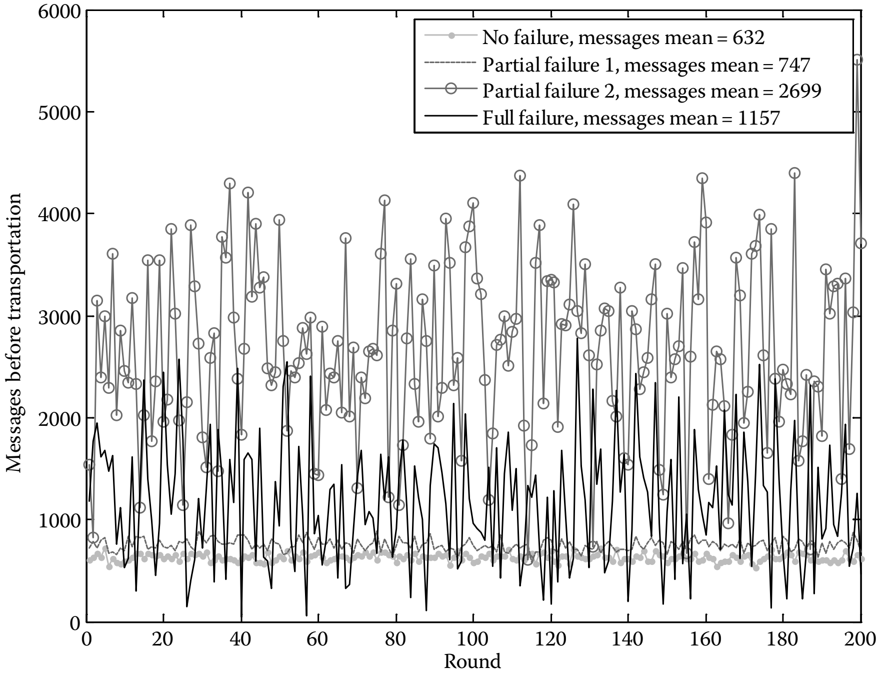


FIGURE 8.11
Communication load due to partial or full failure of facilitating antibody.

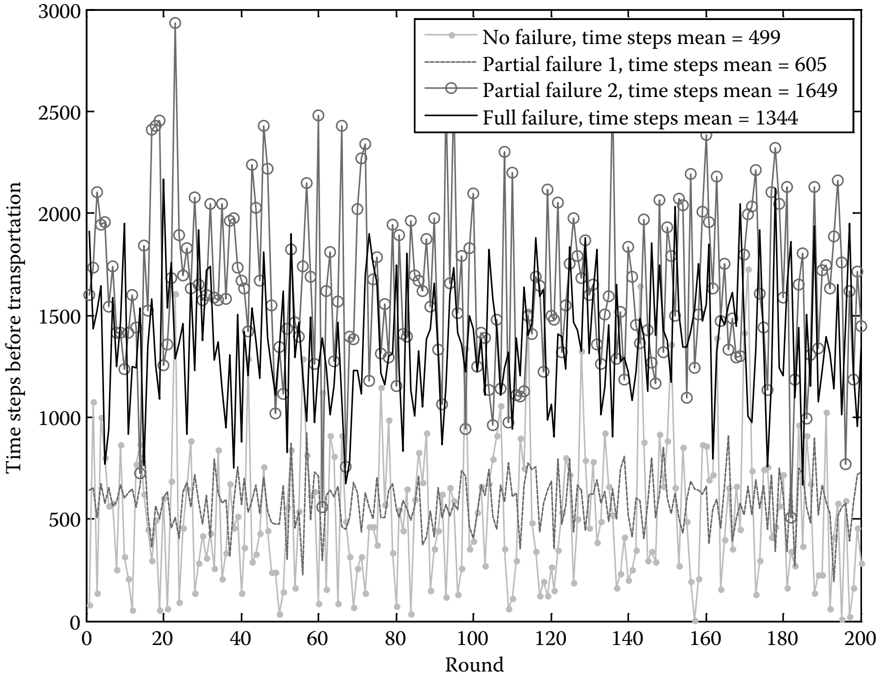


FIGURE 8.12
Effect on time (step) from partial or full failure of instigating antibody.

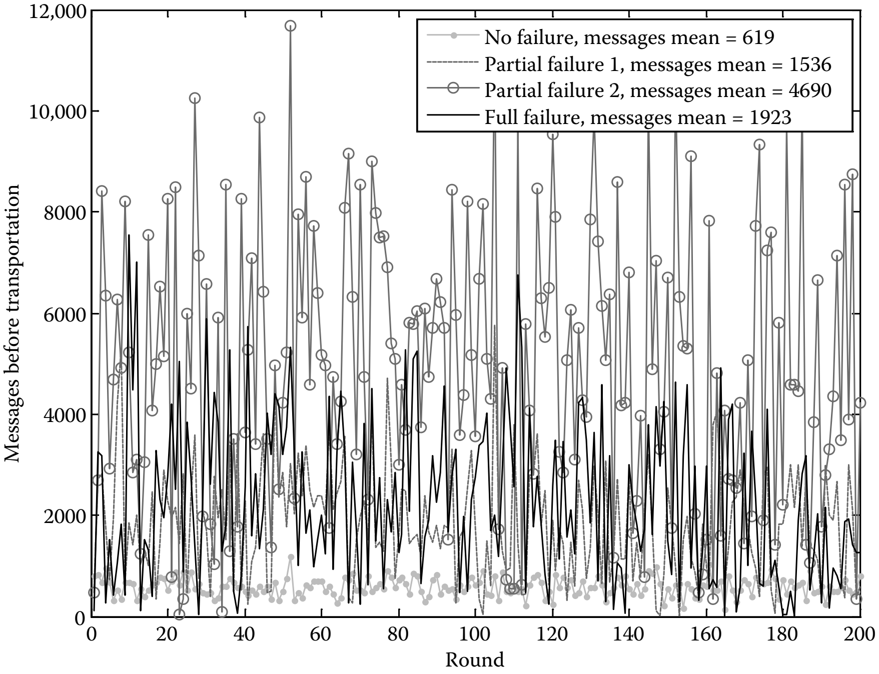


FIGURE 8.13
Communication load due to partial or full failure of instigating antibody.

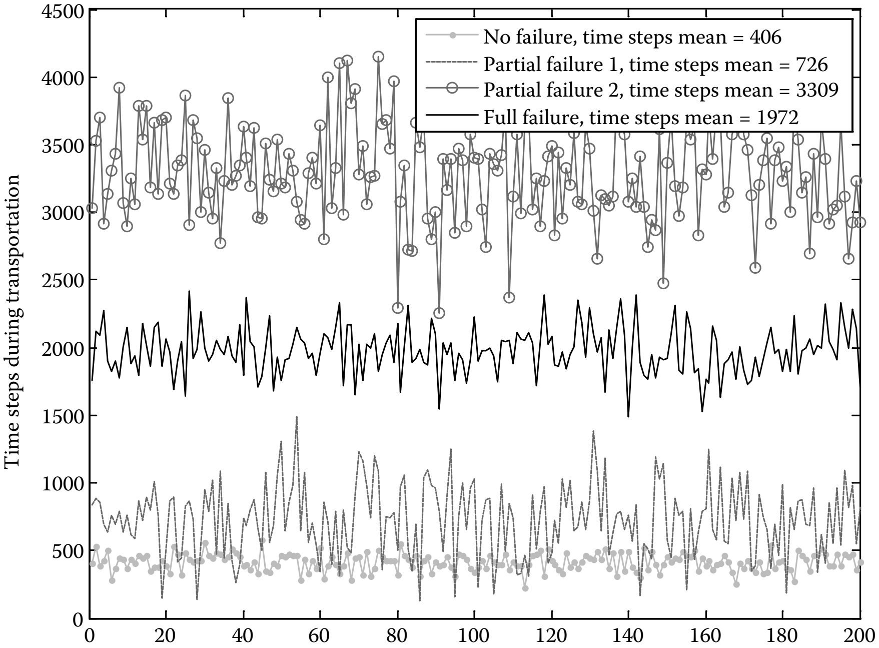


FIGURE 8.14
Effect on time (steps) from partial or full failure of a cooperating antibody during elimination of an antigen.

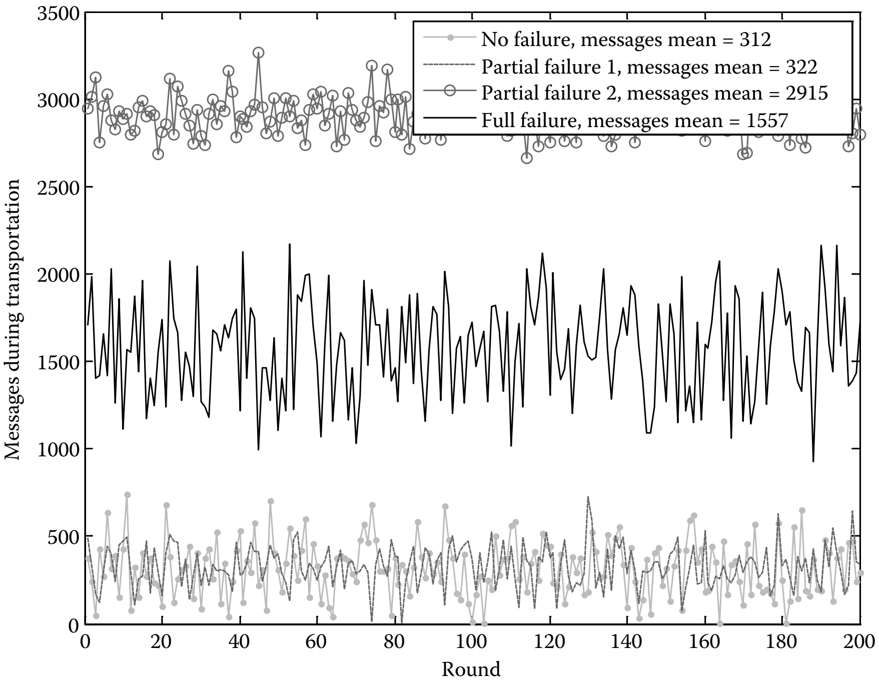


FIGURE 8.15
Communication burden due to partial or full failure of a cooperating antibody during elimination of an antigen.

Figure 8.11 shows the results of communication load on the system due to partial and full failure of the facilitating antibody. Here, communication messages include the broadcast message sent by the instigating antibody to ask for help, the messages sent among other antibodies to determine the antibody having the highest binding affinity, the messages sent by the facilitating antibody to the instigating antibody before and after failure, and additional messages pertaining to selection of a healthy facilitating antibody once the present antibody is declared as failed. As shown in the figure, the communication load increases with failure. However, the system manages to handle all the failures at the cost of extra time (steps) and communication load.

Figures 8.12 and 8.13 show the results of failure of the instigating antibody during coordination before elimination. Here it is important to mention that in order to lower the communication load on the system, only one-way communication is present between the instigating and the facilitating antibodies. The facilitating antibody sends messages periodically until it reaches the destination, and the instigating antibody receives those messages and is reassured that help is forthcoming. During the trek of the facilitating antibody toward an antigen, it does not have any information about the health of the instigating antibody. Once it reaches the antigen, the facilitating antibody sends a “Synchronize” signal to the instigating antibody to start the elimination process. If a reply or the expected response is not received, the facilitating antibody assumes either the instigating antibody has failed or that two antibodies are not adequate to purge the antigen. In either case, the facilitating antibody becomes the initiating antibody and broadcasts a help signal again. Figure 8.12 depicts the effect on time (steps) from the failure of the instigating antibody. It is apparent from this result that in the absence of any malfunction the instigating and the facilitating antibodies utilize the least amount of time (steps) to reach the antigen in order to start the cooperative elimination. In case of Partial failure 1, the time (steps) taken is less than that for both Partial failure 2 and full failure as the instigating antibody is able to communicate its breakdown to the facilitating antibody. The time (steps) taken is the largest in the case of Partial failure 2. This is true because the communication channel of the failed antibody is operational, and it can communicate with the facilitating antibody, which deceives the facilitating antibody. Due to this charade, the facilitating antibody will take more time to reach the stimulation threshold level before declaring the failure of the instigating antibody. Then the facilitating antibody becomes the instigating antibody and broadcasts the help signal again to replace the dead antibody.

Figure 8.13 shows the communication load on the system due to partial and full failures of the instigating antibody. Here, the communication messages include the broadcast message sent by the instigating antibody to seek help, the messages sent by other capable antibodies to each other to find the antibody that has the highest binding affinity, the periodic messages sent by the facilitating antibody to the instigating antibody until it reaches the antigen that needs to be cooperatively eliminated, and the antibody communication to choose another antibody in order to replace a failed instigating antibody. When there is no failure, the messages are at a minimum. Due to Partial failure 1, Partial failure 2, or full failure, the number of messages increases as the facilitating antibody continues to send messages to the instigating antibody until the stimulation level reaches the threshold. Once the instigating antibody is declared failed, the facilitating antibody becomes the instigating antibody, and it broadcasts a help signal again. The entire process of selecting a facilitating antibody will start again, which places an extra communication burden on the system.

Figures 8.14 and 8.15 indicate the antibody failure during cooperative elimination (transportation) of an antigen. As shown in Figure 8.14, the antibodies take additional time (steps) to eradicate an antigen due to the malfunction of an antibody during cooperative elimination. This is true because the collaborating antibody takes time to ensure that the other is failed. Moreover, a healthy antibody has to come and replace the malfunctioned antibody, which also takes time.

In Figure 8.15, when there is no failure, the cooperating antibodies have sent 312 messages to each other during elimination of the antigen. These messages include the messages sent to each other when synchronizing the pushing action, avoiding obstacles during elimination, and rotating an antigen. However, the number of messages increases due to a failure for two reasons. First, the collaborator antibody continues to send messages to the failed antibody asking to make a necessary move until the stimulus threshold level is reached. Second, the antibody has to start a new process of selecting a facilitating antibody to replace the failed one, which requires messaging between antibodies as mentioned before. However, the outcome indicates that the system can handle the failure of an antibody during cooperative elimination of an antigen, although it takes more time (steps) and places an additional communication burden on the system.

When an antibody in the explore state locates an antigen and is able to eliminate it alone, the failure can only occur during the elimination process. In this scenario, Partial failure 2 and the full failure are identical as there is no ally antibody that can pronounce the failure. Then, some other antibody in the explore state will attack a stationary antigen and will start eliminating it. In the case of partial failure, the failed antibody can inform other antibodies in the explore state about the antigen. The antibody that is chosen based on its capabilities and binding affinity will approach the antigen to replace the failed antibody and to eliminate the antigen alone.

Figures 8.16 and 8.17 show the results for eliminating an antigen by a single antibody. As shown in Figure 8.16, when there is no failure the antibody takes less time (steps) to eliminate the antigen. However, in full failure it takes more time (steps) in comparison to partial failure as the malfunctioned antibody cannot communicate its failure to the teammates.

Figure 8.17 shows the communication load due to failure of an antibody during the elimination process. It is clear from the figure that no messages are incurred due to full or no failure. This is true because a single antibody is used to eliminate the antigen and there are no cooperative elimination messages. The figure shows the messages due to partial failure only because an antibody is able to communicate its failure to the teammates. The antibodies in the explore state communicate with each other to choose the most suitable antibody to replace the malfunctioned one.

The overshoots seen in the figures are attributed to many factors, such as the following: The antigens in the explore state wander in the workspace causing interference from time to time during elimination of an antigen, and they are perceived by the cooperating antibodies as dynamic obstacles. As the wandering antibody movement is random and unpredictable, it takes increased coordination and messaging between the two cooperating antibodies to avoid the resulting obstacle. Sometimes antibodies get stuck in confined spaces, such as corners or between other antigens and antibodies. This requires more messaging between the cooperating antibodies to resolve the issue. Moreover, due to failures, when a malfunctioned antibody is replaced by a healthy one, extra messaging will be required between the antibodies to choose the most appropriate antibody. The time (steps) taken by the healthy antibody to reach the antigen increases as well as indicated by the overshoots in the figures.

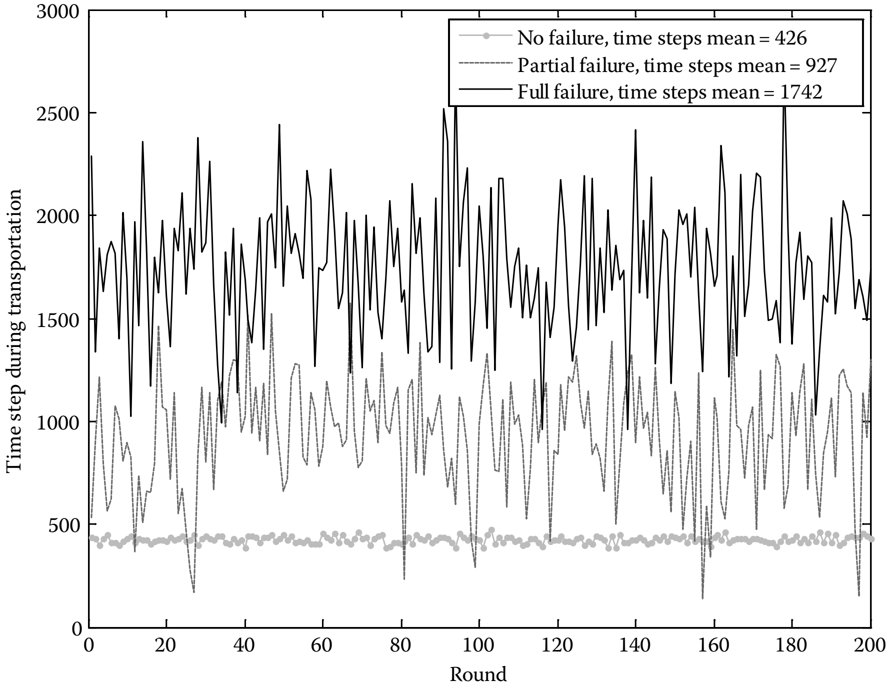


FIGURE 8.16
Effect on execution time (steps) from partial or full failure of an antibody during the elimination of an antigen by a single antibody.

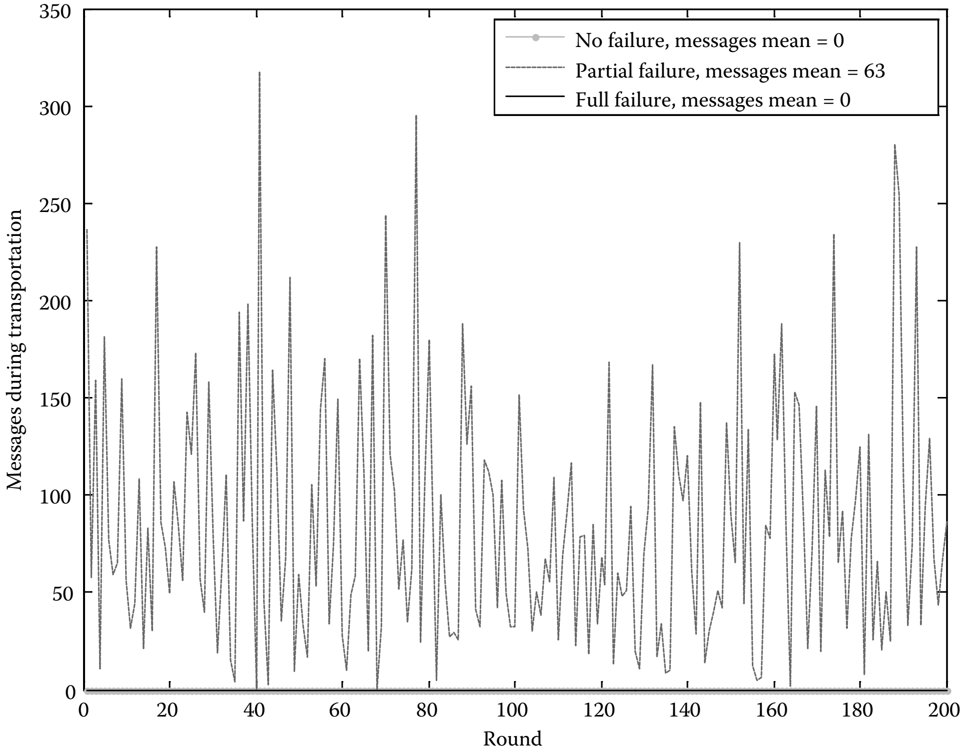


FIGURE 8.17
Communication burden due to partial failure of an antibody during the elimination of an antigen by a single antibody.

8.9 Conclusions

In this chapter, an autonomous, fault-tolerant, and intelligent multi-robot system framework for robot cooperation was presented. It was based on an AIS and genetic algorithm and was applied to transport multiple objects by cooperating robots. A computational method based on the modified Farmer's model of idiotypic network theory for simulating stimulation and suppression phenomenon was developed. In this approach, robots are able to make independent decisions and resolve conflicts. In the developed work, cooperation is not planned a priori and employed by the system only when required. Both partial and full failures were introduced into the robots during different stages of task execution to validate the robustness of the methodology. The developed methodology was implemented through simulation in a team of heterogeneous robots performing a task of multiple object transportation. The results showed that the developed approach was able to successfully complete the desired task in an unknown environment with dynamic and static obstacles.

References

1. Khan, M. T. and de Silva, C. W., "Multi-robot cooperation using an immune system model for multi-object transportation," *International Journal of Robotics and Automation*, Vol. 28, No. 1, pp. 42–56, 2013.
2. Huntsberger, T., Pirjanian, P., Trebi-Ollennu, A., Nayer, H. D., Ganino, A. J., Garrett, M., Joshi, S. S., and Schenker, P. S., "CAMPOUT: A control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration," *IEEE Transactions on System, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 33, No. 5, pp. 550–559, 2003.
3. Murphy, R., Blitch, J. G., and Casper, J. L., "Robocup/AAAI urban search and rescue events: Reality and competition," *AI Magazine*, Vol. 1, No. 23, pp. 37–42, 2003.
4. Parker, L. E., "On the design of behavior-based multi-robot teams," *Advanced Robotics*, Vol. 10, No. 6, pp. 547–548, 1996.
5. Stroupe, A., Huntsberger, T., Okon, A., Aghazarian, H., and Robinson, M., "Behavior-based multi-robot collaboration for autonomous construction tasks," *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005 (IROS 2005)*, Edmonton, Alberta, Canada, pp. 1495–1500, August 2005.
6. McLain, T. W., Rock, S. M., and Lee, M. J., "Experiments in the coordinated control of an underwater arm/vehicle system," *Autonomous Robots*, Vol. 3, Nos. 1–3, pp. 212–232, 1996.
7. Brooks, R. A., "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, Vol. 2, No. 1, pp. 14–23, 1986.
8. Matsumoto, A., Asama, H., Ishida, Y., Ozaki, K., and Endo, I., "Communication in the autonomous and decentralized robot system ACTRESS," *Proc. IEEE Workshop on Intelligent Robots and Systems, IROS*, Ibaraki, Japan, pp. 835–840, July 1990.
9. Asama, H., Ozaki, K., Matsumoto, A., Ishida, Y., and Endo, I., "Development of task assignment system using communication for multiple autonomous robots," *Journal of Robotics and Mechatronics*, Vol. 4, No. 2, pp. 122–127, 1992.
10. Asama, H., Maki, K. H., Endo, I., Ozaki, K., Matsumoto, A., and Ishida, Y., "Functional distribution among multiple mobile robots in an autonomous and decentralized robot system," *IEEE Robotics and Automation Conf.*, Sacramento, CA, pp. 1921–1926, April 9–11, 1991.
11. Jerne, N. K., "Towards a network theory of the immune system," *Annales de l'Institut Pasteur Immunology*, Vol. 125, No. C(1/2), pp. 373–389, 1974.
12. Ronald-Kube, C. and Bonabeau, E., "Cooperative transport by ants and robots," *Robotics and Autonomous Systems*, Vol. 30, No. 1, pp. 85–101, 2000.
13. Lee, M., "Evolution of behaviors in autonomous robot using artificial neural network and genetic algorithm," *Information Sciences*, Vol. 155, Nos. 1–2, pp. 43–60, 2003.
14. Zhao, K. and Wang, J., "Multi-robot cooperation and competition with genetic programming," *Proc. European Conference on Genetic Programming*, Scotland, UK, April 2000.
15. Liu, Z., Ang, Jr., M. H., Khoon, W., and Seah, G., "Multi-robot concurrent learning of fuzzy rules for cooperation," *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, pp. 713–719, June 2005.
16. Farmer, J. D., Packard, N. H., and Perelson, A. S. "The immune system, adaptation, and machine learning," *Physica D: Nonlinear Phenomena*, Vol. 22, No. 1, pp. 187–204, 1986.
17. Bersini, H. and Varela, F. J., "Hints for adaptive problem solving gleaned from immune network," *Proceedings of the Workshop Parallel Problem Solving from Nature*, Dortmung, Germany, pp. 343–354, October 1990.
18. Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. "Self-nonself discrimination in a computer," *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, pp. 202–212, May 1994.

19. Dasgupta, D. and Stephanie, F., *Tool Breakage Detection in Milling Operations Using a Negative-Selection Algorithm*, Technical Report CS95-5, Department of Computer Science, University of New Mexico, Albuquerque, NM, 1995.
20. Dasgupta, D. and Stephanie, F., "Novelty detection in time series data using ideas from immunology," *Proceedings of the International Conference on Intelligent Systems*, Bombay, India, pp. 82–87, December 1996.
21. Dasgupta, D. (Editor), *Artificial Immune Systems and Their Applications*, Springer-Verlag, Inc., Berlin, 1999.
22. Dias, M. B., Zinck, M., Zlot, R., and Stenz, A., "Robust multirobot coordination in dynamic environments," *Proc. of IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, pp. 3435–3442, April–May 2004.
23. Parker, L. E., "ALLIANCE: An architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, pp. 220–240, 1998.
24. Parker, L. E., "Adaptive heterogeneous multi-robot teams," *Neurocomputing*, Vol. 28, No. 1, pp. 75–92, 1999.
25. Gerkey, B. P. and Mataric, M. J., "Pusher-watcher, An approach to fault-tolerant tightly coupled robot coordination," *Proc. IEEE Conf. on Robotic and Automation*, Washington, DC, pp. 464–469, May 11–15, 2002.
26. Gerkey, B. P. and Mataric, M. J., "Sold! Auction methods for multirobot coordination," *IEEE Transactions on Robotic and Automation*, Vol. 18, No. 5, pp. 758–768, 2002.
27. Gao, Y. and Wei, W., "A new multi-robot self-determination cooperation method based on immune agent network," *Proc. IEEE Conf. on Robotics and Automation*, Barcelona, Spain, pp. 390–395, April 2005.
28. Lau, H. Y. K. and Wong, V. W. K., "An immunity-based distributed multiagent-control framework," *IEEE Transactions on System, Man, and Cybernetics- Part A: System and Humans*, Vol. 36, No. 1, pp. 91–108, 2006.
29. Gao, Y. and Luo, Z., "Dynamic task allocation method based on immune system for cooperative robots," *Proc. 7th World Congress of Intelligent Control and Automation*, Chongqing, China, pp. 1015–1020, June 2008.
30. Whitbrook, A., Aickelin, U., and Garibaldi, J., " Idiotypic immune networks in mobile robots control," *IEEE Transactions on Systems, Man, Cybernetics- Part B*, Vol. 37, No. 37, pp. 1581–1598, 2007.
31. Luh, G. and Liu, W., "An immunological approach to mobile robot reactive navigation," *Applied Soft Computing*, Vol. 8, No. 1, pp. 30–45, 2008.
32. Ichikawa, S., Kuboshiki, S., Ishiguro, A., and Uchikawa, Y., "A method of gait coordination of hexapod robots using immune network," *Artificial Life and Robotics*, Vol. 2, No. 1, pp. 19–23, 2006.
33. Wang, Y., You, Z., and Chen, C., "AIN-based action selection mechanism for soccer robot systems," *Journal of Control Science and Engineering*, Vol. 2009, pp. 1–10, 2009.
34. Li, J., Xu, H., Wang, S. and Bai, L., "An immunology-based cooperation approach for autonomous robots," *Proc. Int. Conf. on Intelligent Systems and Knowledge Engineering*, Chengdu, China, pp. 1–6, October 2007.
35. Sathyanath, S. and Sahin, F., "AISIMAM-An AIS based intelligent multiagent model and its application to a mine detection problem," *Proceedings of the ICARIS 2002 1st International Conference on Artificial Immune Systems*, Canterbury, UK, pp. 1–10, September 2002.
36. Sathyanath, S. and Sahin, F., "Application of artificial immune system based intelligent multi agent model to a mine detection problem," *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, Hammamet, Tunisia, pp. 6–9, October 2002.
37. Cai, Z. and Peng, Z., "Cooperative coevolutionary adaptive genetic algorithm in path planning of cooperative multi-robot systems," *Journal of Intelligent and Robotic Systems*, Vol. 33, No. 1, pp. 61–71, 2002.
38. Kubota, N., Morioka, T., Kojima, F., and Fukuda, T., "Learning of mobile robots using perception-based genetic algorithm," *Measurement*, Vol. 29, No. 3, pp. 237–248, 2001.
39. Yang, J. and Luo, Z., "Coalition formation mechanism in multi-agent systems based on genetic algorithms," *Applied Soft Computing*, Vol. 7, No. 2, pp. 561–568, 2007.

Section II

Applications

9

Automated Mechatronic Design Tool

Saeed Behbahani, Leila Pezeshki, and Clarence W. de Silva

CONTENTS

9.1 Introduction 324
  9.1.1 Mechatronic Design Theory 324
9.2 Evolutionary Mechatronic Tool 325
  9.2.1 Genetic Programming 326
  9.2.2 Bond Graphs 327
  9.2.3 Integration of Bond Graphs and Genetic Programming 328
    9.2.3.1 Bond Graph Embryo Model 328
    9.2.3.2 Construction Functions and Terminals 329
    9.2.3.3 First Generation 330
    9.2.3.4 Creation of New Generations 332
    9.2.3.5 Case Study: Eigenvalue Assignment 333
9.3 Controller Design Using Bond Graphs 335
9.4 Two-Loop Design Model 337
  9.4.1 Hybrid Genetic Algorithm with Genetic Programming 339
  9.4.2 Case Study: Iron Butcher Controller Design 341
9.5 Niching Optimization Scheme 345
  9.5.1 Niching Genetic Programming 346
  9.5.2 Case Study: Model-Referenced Active Car Suspension 347
  9.5.3 Case Study: Hydraulic Engine Mount Design 349
9.6 Conclusions 356
References 356

SUMMARY This chapter presents a theory of concurrent mechatronic design and the development steps of an evolutionary mechatronic design tool. The aim is to synthesize the structure of a mechatronic design solution from a very basic and high-level definition of the problem as given by an expert designer. The automated design tool is generated by integrating a bond graph-based modeling tool as an integrated modeling environment for mixed mechatronic systems, together with genetic programming as a powerful topology exploration and optimization tool. Advanced features of the tool are presented by using a two-loop design model as well as by incorporating a niching optimization scheme. An ultimate goal of this approach is to automate the intelligence that is provided by a human expert designer in evolving an optimal design for an engineering system.

9.1 Introduction

Mechatronics refers to those systems, devices, and machines with which there exists a synergic integration and interaction between mechanical, electrical, computer, and control systems [1]. The general concept is shown in Figure 9.1. This term was initially introduced to the engineering literature in the late 1960s, but its significant growth can be traced back to the 1990s, in relation to the "mechatronic" development of electromechanical systems. It is popularly used in such engineering applications as machine tools, biomechanics, robotics, automobiles, aerospace systems, food processing machinery, and home appliances.

Because there exists an ongoing pursuit to offer better mechatronic products at lower cost, today there is significant interest and activity in the area of optimal design of mechatronic systems.

9.1.1 Mechatronic Design Theory

In any engineering design problem, the objectives are to synthesize the required system and to realize design parameters with the aim to achieve the required behavior without violating a specified set of constraints. Alternatively, the design problem can be viewed as a mapping from a required behavior space to a parameter space.

In a traditional design process, termed electromechanical design, the mechanical structure of the system is designed and constructed first. Next, the electrical components (e.g., actuators, sensors) are added based on such requirements as maximum required acceleration and minimum acceptable accuracy. After the machine is built, it would be costly to vary a parameter in the mechanical structure or change electrical hardware; hence, these parameters can be treated as time invariant. Subsequently, the controller parameters that are tunable and programmable (e.g., control algorithm, control gain, signal conditioning algorithm) are determined based on changeable specifications (e.g., desired path, speed, accuracy, stability).

In such a traditional design methodology, different subsystems are designed and optimized with regard to the particular criterion that is important for the particular subsystem.

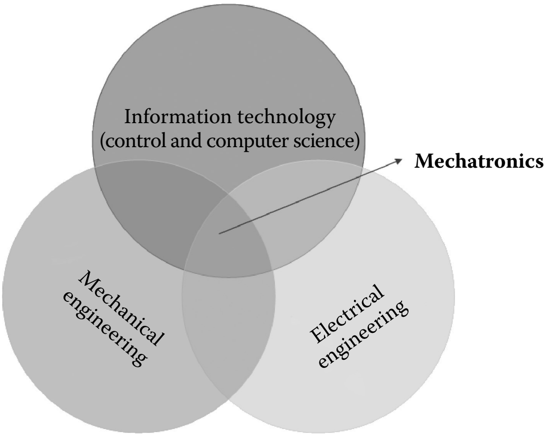


FIGURE 9.1
A general description of mechatronics.

In fact, advancements in control engineering and computer science have resulted in a mindset that the design of the hardware of the system (i.e., mechanical structure and electrical hardware) is no longer the main design focus, and the inadequacies of the system could be compensated for by sophisticated control schemes and software. This thinking may not always be correct, in particular for a complex system with considerable dynamic interactions between its subsystems. In view of hardware limitations and dynamic interactions, a perfect and optimum system performance may be hardly achievable, regardless of the effort devoted to the control design. It can be argued that a machine's performance can be enhanced by a proper control scheme, but the "adequacy" and the "optimality" of the system that is designed through such a sequential process are questionable [1].

In view of dynamic interactions, optimal mechatronic design needs integrated, concurrent, and system-based methodology. In this school of thought, it is argued that the controllability and programmability of control parameters should be considered as an opportunity to further improve and tune the performance of a mechatronic system because it is quite likely that the required specifications are not exactly matched with the results of the design process.

An engineering design is always a multi-criteria problem. In view of the presence of different subsystems in a mechatronic system, there are some particular criteria related to each subsystem and the required criteria of the overall system. Due to dynamic interactions, the components of the system may affect the objectives of other components. Ideal mechatronic design has to follow a system-based method, meaning that it has to improve the product performance from different views considered simultaneously—to improve the overall degree of satisfaction of the system.

9.2 Evolutionary Mechatronic Tool

In a typical mechatronic design problem, the designer needs to synthesize the optimum structure (i.e., topology and size) of a machine that properly meets the requirements of the machine. Topology refers to the number of components, their type, and the way they are interconnected together, and sizing refers to the numerical values of the component parameters [2]. The complexity of the mechatronic design problem usually comes from the fact that the topology of the best solution has an unlimited growth capability and uncertainty, which has to be resolved by an expert designer by devoting significant intelligence, creativity, domain knowledge, and experience [1]. Also, the topology offered by expert designers for a system is evolved gradually, meaning that there is no guarantee that the offered structure is the absolute optimal solution.

Considering the open-ended nature of topology search space and its dynamic nature, the main challenge in creating a tool for autonomous design of mechatronic systems is to generate a method to effectively explore all possible topologies for a mechatronic system.

Genetic programming (GP) is a branch of evolutionary algorithms that has open-ended growth possibility for optimal solutions, which makes it an appropriate option for topology exploration and optimization if it is integrated with a suitable modeling method [2–4]. Because a mechatronic system contains components from different engineering domains, we need a modeling method to represent components from different domains in a unified language. Bond graph (BG) modeling, which has a domain-free graphical representation, meets the requirement of integrated design for mechatronic systems [5–8].

9.2.1 Genetic Programming

Evolutionary algorithms and in particular genetic algorithms are optimization methodologies inspired by the natural evolution of biological species. They are stochastic search methods based on Darwin's principle of survival of the fittest. There are many different branches of genetic algorithms (GAs), but the essence of all of them is nearly the same. A genetic algorithm has five main steps (Figure 9.2):

1. Genetic operation is tuned by determining the main characteristics of the procedure, such as the population size and the probability of each reproduction operation (crossover, mutation, and survival).
2. The first population of solutions is generated randomly.
3. Each individual solution is scored by considering a predefined fitness function to evaluate how well it satisfies the desired specifications. In this step, the GA may have to be linked to a simulation tool for analyzing the behavior of the trial solutions.
4. A new population of solutions is created by imitating the evolution of species. New generation is formed based on Darwin's principle of survival of the fittest through the reproduction operations such as mutation, crossover, and survival (selection).

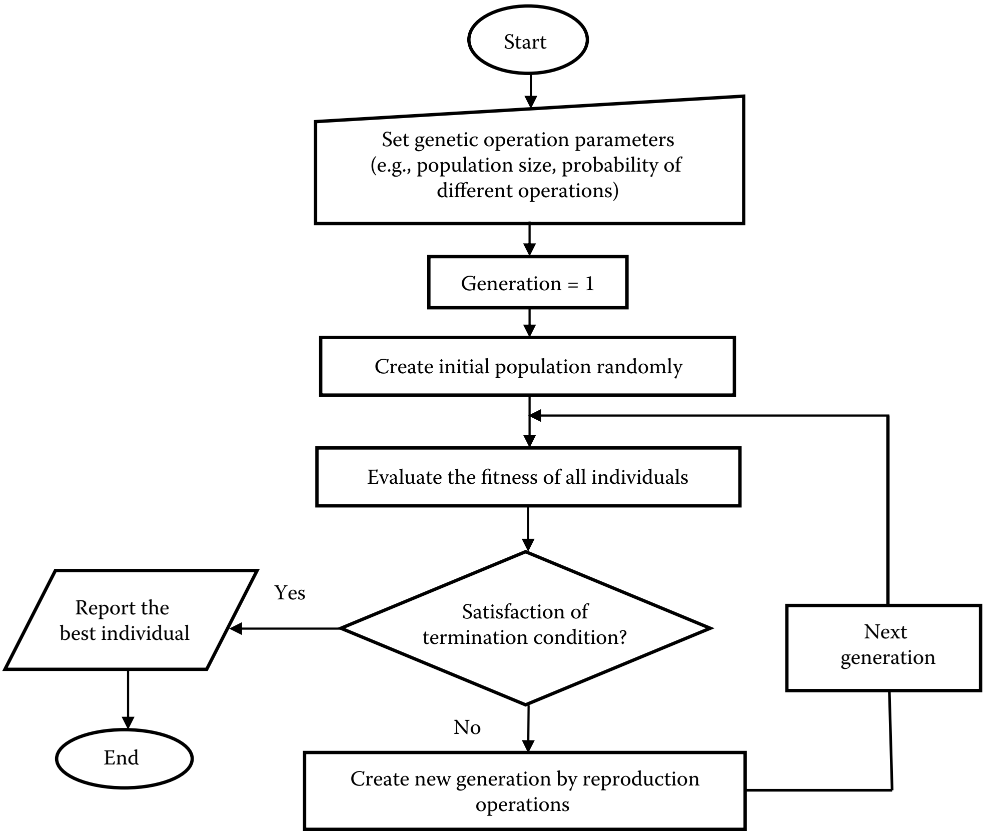

Start
Set genetic operation parameters (e.g., population size, probability of different operations)
Generation = 1
Create initial population randomly
Evaluate the fitness of all individuals
Satisfaction of termination condition?
Yes
No
Report the best individual
End
Create new generation by reproduction operations
Next generation

FIGURE 9.2
Flowchart of genetic algorithm.

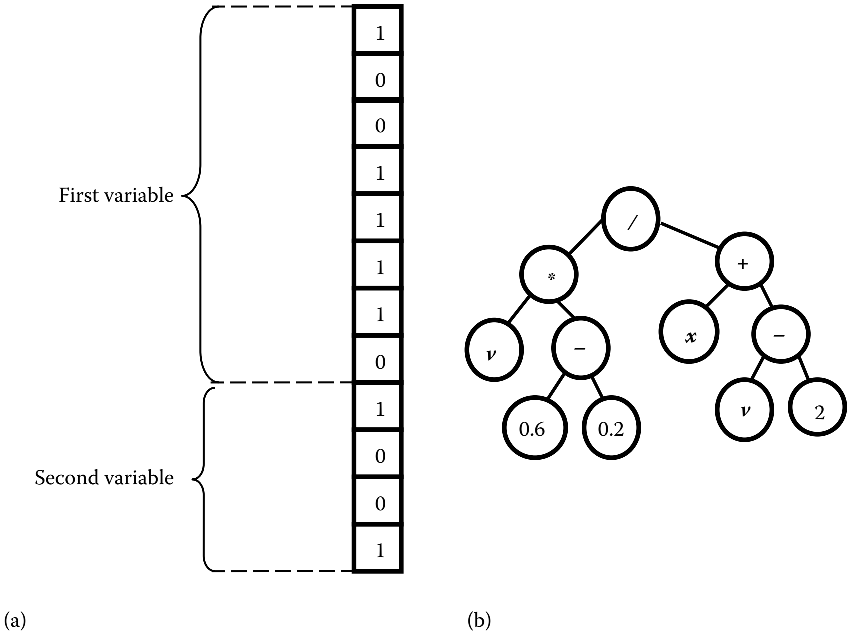


FIGURE 9.3
Chromosome-like representation of individual solutions. (a) Conventional GA. (b) Tree-like representation in GP.

5. The evolution is continued by returning to step 2 until the satisfaction condition is met or a limit on the number of iterations is exceeded. The best-so-far solution is designated as the outcome of the genetic algorithm.

Genetic programming (GP) is a branch of evolutionary computing, which has the same procedure as outlined in Figure 9.2. The main difference between GP and conventional GAs comes from the particular representation of the solution [9]. In a conventional GA, each individual (trial) solution is represented as a series of zeros and ones (Figure 9.3a) whereas in GP a tree formed by *construction functions* represents a solution (Figure 9.3b). Each tree-like combination of construction functions creates an individual solution when the functions are applied to a predefined *embryo* model. These concepts are explained in Section 9.2.3. GP and its tree-like representation were initially utilized in exploring and evolving mathematical equations [3] as shown in Figure 9.3b. The other features of GP are by and large identical with genetic algorithms, such as reproduction operations and determining appropriate fitness evaluation methodology. Further information about the genetic programming procedure and its components can be found in [3,9].

9.2.2 Bond Graphs

A mechatronic system needs a modeling tool that can handle different components belonging to different domains, such as mechanical, electrical, hydraulic, thermal, and control in a unified and integrated environment [1]. The most commonly used modeling or simulation tools are appropriate only for a single specific domain although in a multidomain mechatronic system, due to the component interactions, it is required to simulate and analyze the entire system in an integrated manner. Bond graphs (BGs) are proven to be an effective modeling method for mixed systems [6–8]. They have a unified graphical representation for lumped systems, in which a combination of bond and node components can be used to generate any system model, rather than by the specification of equations. This graphical representation of BGs also has an open architecture. This means that a free composition of bonds and nodes can be added to different locations of a model to create a new model with a new topology. This unbounded growth ability of BGs provides the capability to explore a wide range of topologies in a process of design and optimization. In other words, the use of BGs for mechatronic design can result in an optimization tool that will not be restricted to the system sizing aspect but will be applicable to optimizing the topology of the model as well.

The final product of a bond graph model is the state space dynamic equations of the system. These equations can be solved or simulated in the time domain by using numerical methods or they can be used to find the transfer function of the system. Accordingly, the state space equations can be used to design a suitable controller for the system. It also leads to derivation of the equations and simulation of the entire system by considering the added controller. A detailed description of the BG components and characteristics and the procedure of modeling different systems using this tool can be found in [1,8].

NOTE: An alternative modeling tool to bond graphs is linear graphs (LGs). An introduction to LGs is found in [1].

9.2.3 Integration of Bond Graphs and Genetic Programming

The graphical flexibility of BGs and the extensive search ability of GP have provided the rationale for integration of these two powerful tools. The tree-like representation of GP and its unlimited growth capability make GP suitable for optimization of the structure of the solution for a particular problem (i.e., both topology and sizing) if a mapping is provided between each tree-like representation and an individual solution in the search space of that problem. Two main concepts in providing this mapping are the embryo model and the construction functions.

9.2.3.1 Bond Graph Embryo Model

The embryo is a basic initial model given by the designer for the problem to be solved. It includes a very basic structure of the solution and remains unchanged in all alternative solutions. In fact, the embryo is a structure that the final solution is supposed to be constructed over. It is problem-dependent, and it is the user who should decide the level of details in the embryo. As a general rule, the inputs and outputs of the problem, which are involved in the fitness evaluation, should be available in the embryo model. The designer needs to introduce several *modifiable sites* on the embryo, which can be improved by applying construction functions. Considering the typical BG embryo model shown in Figure 9.4, three types of modifiable sites can be observed [4]:

1. Modifiable joints, shown by dashed circle
2. Modifiable bonds, shown by dashed square
3. Arithmetic sites, shown by dashed elliptic

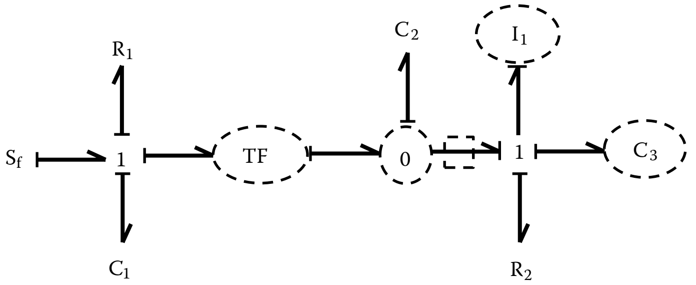


FIGURE 9.4
Typical hydraulic engine mount as a sample BG embryo model.

9.2.3.2 Construction Functions and Terminals

Construction functions are the alphabet of creation of new models. They are functions that can be applied to the embryo in order to create new and extended models. They have an open architecture. This means that they can create new modifiable sites in the model. This feature makes them grow like branches of a tree. Proper termination functions are also necessary to prevent unlimited growth of branches.

Each of the already-explained modifiable sites has its own possible construction functions. Now, the construction functions developed in [4] for linear systems are explained.

9.2.3.2.1 Add-Element Functions

Consider a modifiable node. A lumped one-port element (i.e., resistor, capacitor, or inductor) may be added to this node [4,10], or it may be kept without any extension. Figure 9.5 shows a modifiable node before and after adding a one-port element, a resistor in this case. When the new element is added, first the old modifiable site should be kept modifiable for further modifications. Second, the added element needs a numeric value, which is provided by an arithmetic modifiable site. In addition, the bond that connects the element to the initial node is also a site at which the model can be further extended. It means that when an add-element function is applied to a modifiable node, two new modifiable sites are created, and the old modifiable site is also kept for further modification (Figure 9.5). Consequently, this function has three arguments. The first one indicates the value of the added element with arithmetic functions; the second one is the old modifiable site for

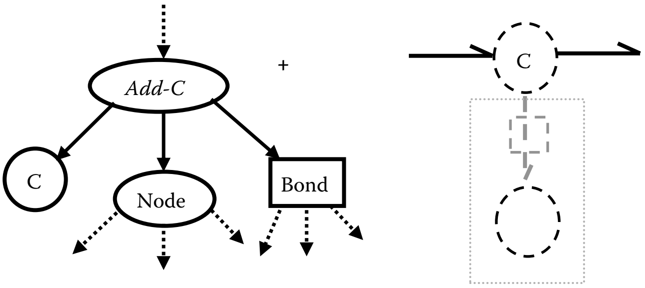


FIGURE 9.5
Add-element construction function.

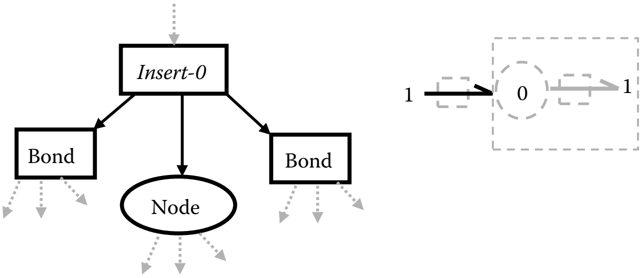


FIGURE 9.6
Insert-junction construction function.

further modification; and the last one is a bond-type site for inserting a junction at the connection point of the added bond. There are three types of add-element functions: *add-R*, *add-C*, and *add-I*. As these functions may give rise to an unending sequence of add operations, it is necessary to have the *end-node* function to terminate the operation.

9.2.3.2.2 Insert-Junction Functions

Consider now a modifiable bond. In a random base, a junction may be inserted between bonds, or it may be kept without further extension [4,10]. This function creates three new modifiable sites as shown in Figure 9.6. As these functions have a recursive operation, it is necessary to give sufficient probability to a termination operation for them. Accordingly, for each modifiable bond, three options are available to be chosen randomly: *insert-0, insert-1*, and *end-bond*.

9.2.3.2.3 Arithmetic Functions

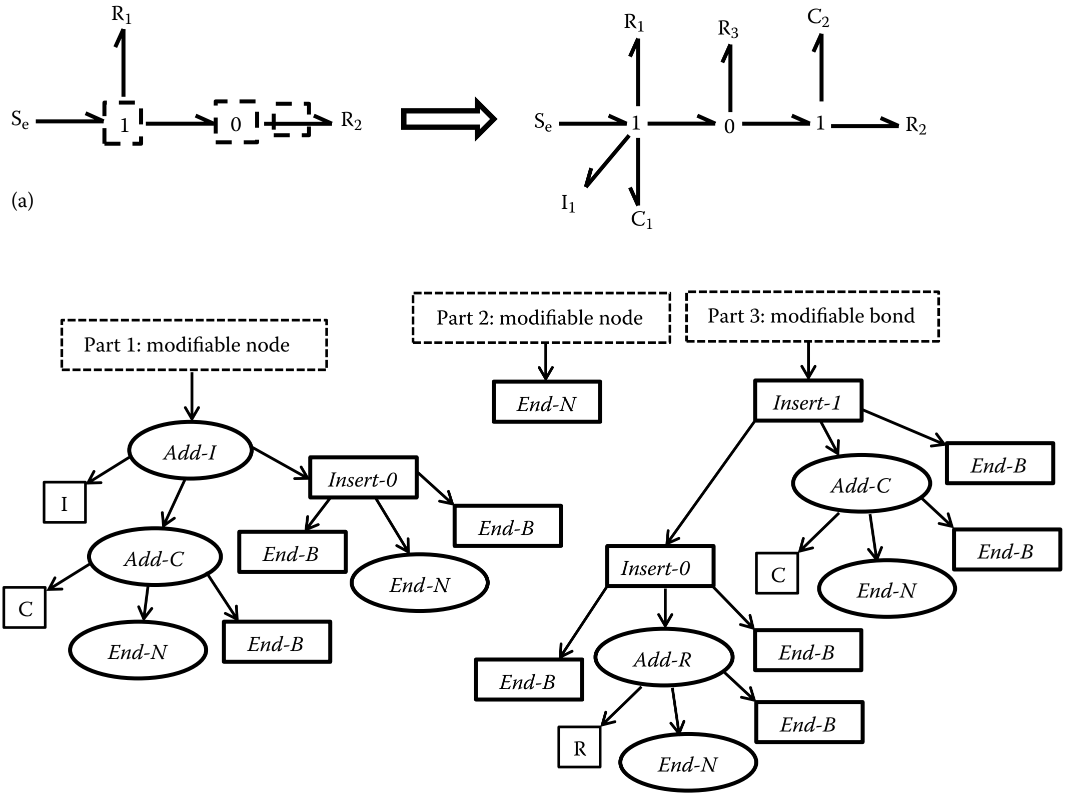
Each added new element or a modifiable element in the embryo model needs a value, which is provided by an arithmetic modifiable site. For each arithmetic modifiable site, three options are available to be chosen randomly: addition, subtraction, and a random number [4]. The first two options open two new modifiable sites (i.e., they have recursive operation), and the third option plays the role of the operation termination, which has to be given sufficient probability to prevent unlimited growth of the tree. Regardless, the result of an arithmetic subtree will be a number, which represents the numerical value of the element. The resulting number may be interpreted in various ways, depending on the designer. For example, it may be interpreted as the logarithmic value of an element [4] or may be interpreted as the change in an initial default value [10]. The value of parameter ζ is calculated by

$$\zeta = \zeta_d (1 + \rho.\varepsilon) \tag{9.1}$$

Here ε is the result of the arithmetic subtree, ρ is a modification factor, and ζ_d is the default value for parameter ζ. It is found desirable to apply a rather high ρ in the beginning of a GP operation for coarse modification and then reduce it later for fine modification.

9.2.3.3 First Generation

The first generation of a population is created by applying random combinations of construction functions to the embryo. For each modifiable site in the embryo, a random number is created by the program. Depending on the value of the random number and the predefined ranges for each available function, one of the available functions is selected and is applied on the modifiable site. For example, if there is a modifiable node in the embryo model, one of the *add-R, add-C, add-I,* or *end-node* functions is selected based on the value of the randomly generated number and is applied to the modifiable node. If *end-node* is selected, then the process for this modifiable site is complete, and the program can go to the next modifiable site. If one of the other options is selected, then the applied function creates some inner modifiable sites. This reproduction can continue like branches of a tree. However, the probability for *end-node* should be sufficiently high to prevent an infinite loop [10]. Figure 9.7 shows an example of the way a tree-like composition of construction functions is stored and the result of its application on the embryo model.


R1
Se
1
0
R2
(a)
R1
R3
C2
Se
1
0
1
R2
I1
C1
Part 1: modifiable node
Part 2: modifiable node
Part 3: modifiable bond
Add-I
I
Insert-0
End-B
Add-C
C
End-B
End-N
End-N
End-B
End-N
Insert-1
End-B
Add-C
End-B
C
End-N
Insert-0
End-B
Add-R
End-B
R
End-N
End-B

S_1 = (*Add-I, I, Add-C, C, End-N, End-B, Insert-0, End-B, End-N, End-B*)

S_2 = (*End-N*)

S_3 = (*Insert-1, Insert-0, End-B, Add-R, R, End-N, End-B, End-B, Add-C, C, End-N, End-B, End-B*)

Construction function string = (S_1, S_2, S_3)

(b)

FIGURE 9.7
An example of creating a new topology using the integration of BG and GP. (a) Embryo model and the resulting BG model from adding the construction functions. (b) Tree-like structure of construction functions and their corresponding string.

9.2.3.4 Creation of New Generations

The tree-like structure of the solutions in the integration of BGs and GP fits the requirements for use in a genetic algorithm. Specifically, the genetic reproduction operations, such as mutation, crossover, and survival (selection), can be used to create new structures for the next generation. The most important of these, which has the highest percentage of reproduction probability, is the crossover operation. In the crossover operation, two tree-like structures are mated to form two new solutions or offspring. The parents are randomly selected from the population but in such a way that the members with higher fitness have a greater chance to be selected. The creation of the offspring from the crossover operation is accomplished by exchanging a branch between parents (Figure 9.8).

Mutation is another operation of genetic programming. In mutation, a tree-like structure is selected randomly but in such a way that the solutions with higher fitness have a greater chance to be selected. In this operation a branch is selected randomly and replaced by a new randomly created branch in order to create a new BG model as shown in Figure 9.9. Mutation gives a very important strength to genetic programming. For example, in the hill climbing method of optimization, the solution may get locked into a local optimum and be unable to escape it. The mutation operation of genetic programming avoids or at least reduces this local minimum problem. It works like a gunshot on the optimization process [11].

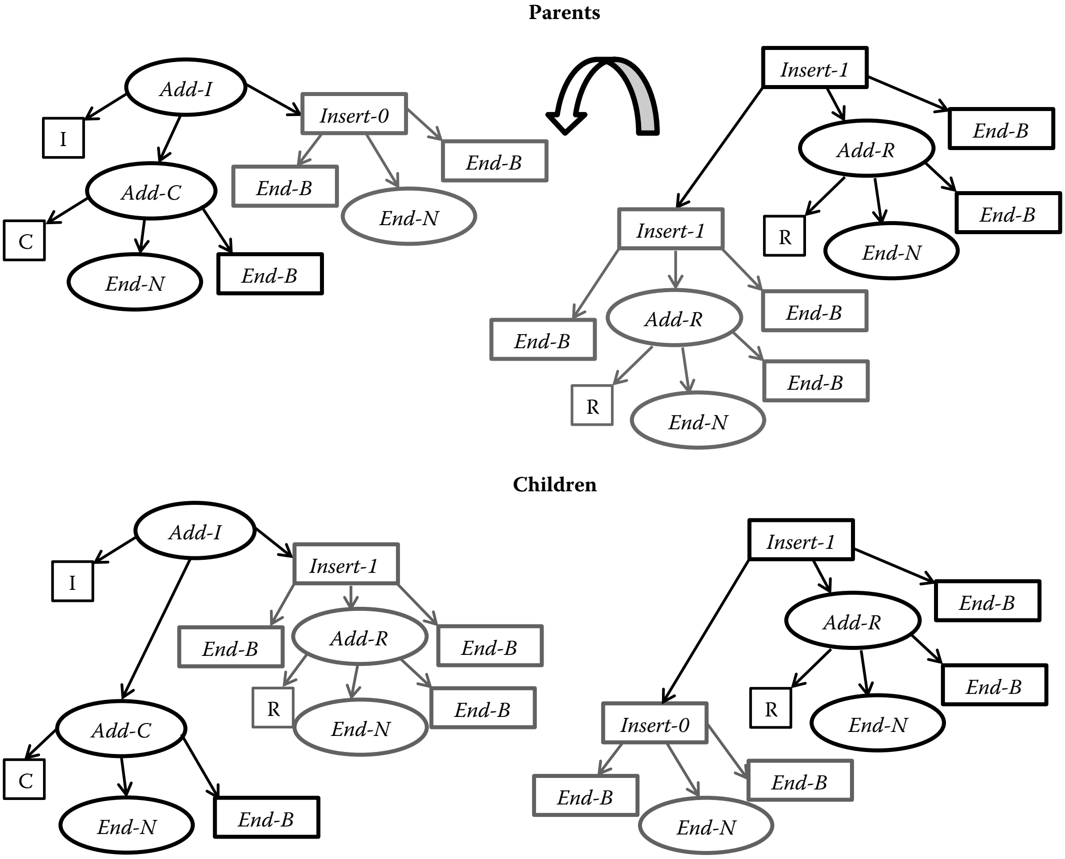

Parents
Add-I
I
Add-C
C
End-N
End-B
Insert-0
End-B
End-N
End-B
Insert-1
Add-R
R
End-N
End-B
End-B
End-B
Insert-1
End-B
Add-R
R
End-N
End-B
End-B
Children
Add-I
I
Insert-1
End-B
Add-R
R
End-N
End-B
End-B
Add-C
C
End-N
End-B
Insert-1
Add-R
R
End-N
End-B
End-B
Insert-0
End-B
End-N
End-B

FIGURE 9.8
Crossover reproduction operation on tree-like BG–GP representation.

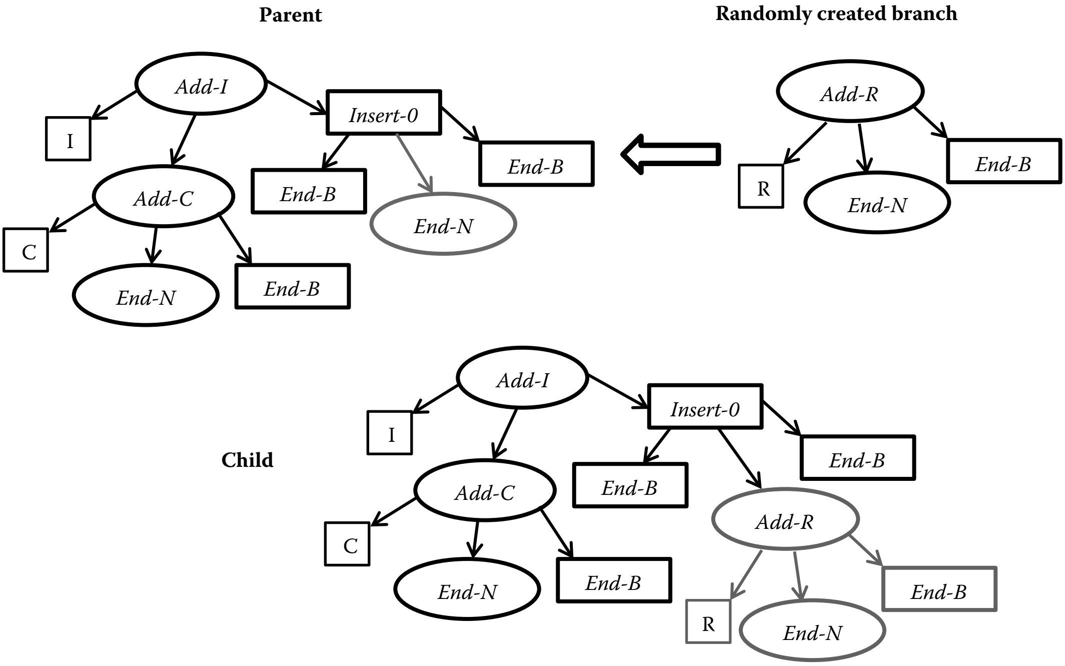


FIGURE 9.9
Mutation reproduction operation in tree-like representation of BG–GP.

In addition to the abovementioned operations, there is a critical operation in which the best solutions are always copied to the next generation in order to guarantee that the best solutions are retained. This operation also guarantees that no degradation will happen in the best solutions of the population and, in turn, guarantees the convergence of the optimization process. Therefore, a predefined number of tree-like structures with the best finesses are transferred to the next generation without any change.

9.2.3.5 Case Study: Eigenvalue Assignment [4]

Now, a simple example is presented to demonstrate the utility of the integrated optimization tool. The eigenvalue assignment is an important concept in the controller design, which provides desirable response characteristics while avoiding instability. In this example, a set of eigenvalues is considered as the target, and the purpose is to find the bond graph model that has the minimal distance errors from the desirable values. The set of target eigenvalues is as follows:

$$\text{Target set} = \{-1 \pm 2i, -2 \pm 1i\}$$

In order to define the fitness function, each target eigenvalue is paired with the closest one in the solution, and the error between each pair is calculated. The sum of the calculated errors is divided by the order, and finally, the fitness function is defined by performing hyperbolic scaling as follows:

$$\text{Fitness function} = 0.5 + \frac{1}{\left(2 + \frac{\sum \text{error}}{\text{order}}\right)} \tag{9.2}$$

Two types of embryo models are used for this problem with different numbers of modifiable sites. Figure 9.10a shows the embryo model with one modifiable site (dashed squares), and the other embryo model with three modifiable sites is presented in Figure 9.10b. Fixed elements in the embryo model are used to define the input and output of the system. The parameter values of the two impedances are presented in Figure 9.10.

The BG models of the solutions are presented in Figure 9.11. In [4], the optimization process has been performed 10 times for each case, and the resulting solutions are compared statistically. Table 9.1 gives the solution eigenvalues and the average distance error from the corresponding target values.

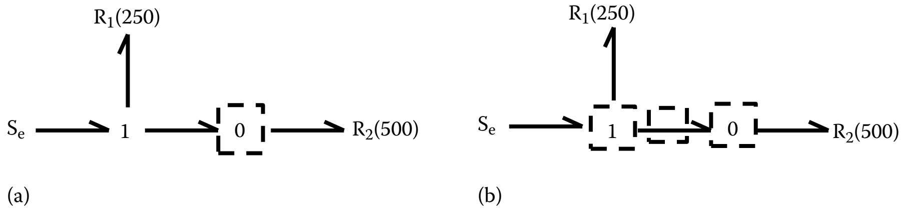

$R_1(250)$
S_e
1
0
$R_2(500)$
(a)
$R_1(250)$
S_e
1
0
$R_2(500)$
(b)

FIGURE 9.10
Two types of embryo models for eigenvalue assignment with (a) one modifiable site and (b) three modifiable sites.

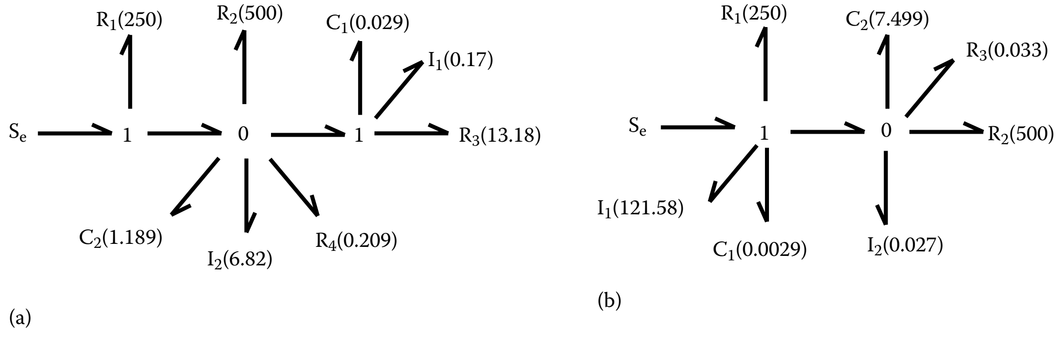

$R_1(250)$
$R_2(500)$
$C_1(0.029)$
$I_1(0.17)$
S_e
1
0
1
$R_3(13.18)$
$C_2(1.189)$
$I_2(6.82)$
$R_4(0.209)$
(a)
$R_1(250)$
$C_2(7.499)$
$R_3(0.033)$
S_e
1
0
$R_2(500)$
$I_1(121.58)$
$C_1(0.0029)$
$I_2(0.027)$
(b)

FIGURE 9.11
BG model of the solution eigenvalues resulted from embryo with (a) one modifiable site and (b) three modifiable sites.

TABLE 9.1
Solution Eigenvalues and Average Distance Error

Modifiable Sites in Embryo	Solution Eigenvalues	Average Distance Error
One	$-0.996 \pm 1.997i, -1.989 \pm 1.001i$	0.008
Three	$-1.028 \pm 1.929i, -2.008 \pm 0.995i$	0.043

9.3 Controller Design Using Bond Graphs

It can be said that a mechatronic system is an integration of physical systems and control systems. Because these two parts have different natures, different tools are required for their modeling. BG modeling, as explained in Section 9.2.2, handles the modeling of the physical subsystem. In order to incorporate the controller subsystem and unify the model, a technique called "controller design in the physical domain" [12,13] may be used, in which the controller is represented by an equivalent physical system. In this method, various combinations of bond graph elements are used to generate different control schemes.

Control methodologies can be categorized as collocated control and noncollocated control. These two categories are determined based on the location of the sensors and actuators. Locating the sensors and actuators at the same place results in collocated control. In this case, the sensors and actuators are energetically conjugated. Collocated control can be represented as an effort-flow one-port pair in bond graph modeling [14] (Figure 9.12).

In the bond graph representation of a collocated controller structure, a sensor and an actuator are connected with a negative feedback loop. For these controllers, the following four cases can be considered [14]:

1. Negative velocity feedback can be considered as a damping action.
2. Negative position feedback can be considered as a spring action.
3. Negative force feedback can be considered as decreasing inertia.
4. Positive force feedback can be considered as increasing inertia.

Physical systems are classified into the two major categories of impedances and admittances, according to the nature of their inputs and outputs. The impedances receive flow

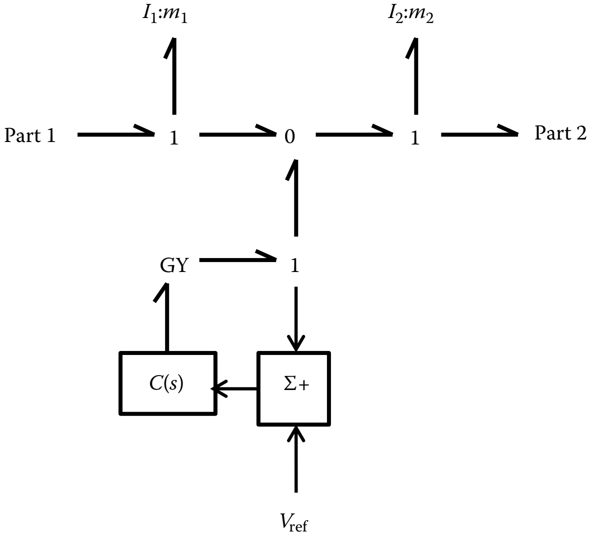


FIGURE 9.12
Collocated control with relative velocity feedback. (From Wang, J., *Integrated Coevolutionary Synthesis of Mechatronic Systems Using Bond Graphs*, PhD Thesis, Industrial Engineering and Operations Research, The University of Massachusetts, Amherst, MA, 2004.)

as input and produce effort outputs, and admittances produce flow output as a function of the received effort input. A passive physical system and an active controller complement each other through their interaction. In other words, when a physical system operates as an admittance, the control system is expected to operate as an impedance. This is equivalent to the concept of causality in bond graph modeling. In a bond graph model, both flow and effort variables exist at a port, and only one of them can be controlled. One of them will be the input to the port, and the other one will be the output from that port. In other words, when two components are connected together by a bond, the effort variable in the bond causes one of the elements to respond with a flow, and this flow causes the other element to respond with an effort. In bond graphs, the direction of the effort signal is shown by a short perpendicular line at the end of a bond and is called a *causality stroke*. For instance, suppose a bond graph model in which one impedance and one admittance are connected by a one-junction. For an integrated system, the admittance can be considered as the physical plant, and the impedance can be determined as a control system. The entire model would represent a closed-loop feedback control system. The bond graph representation of the feedback and feed-forward collocated control structures are shown in Figure 9.13.

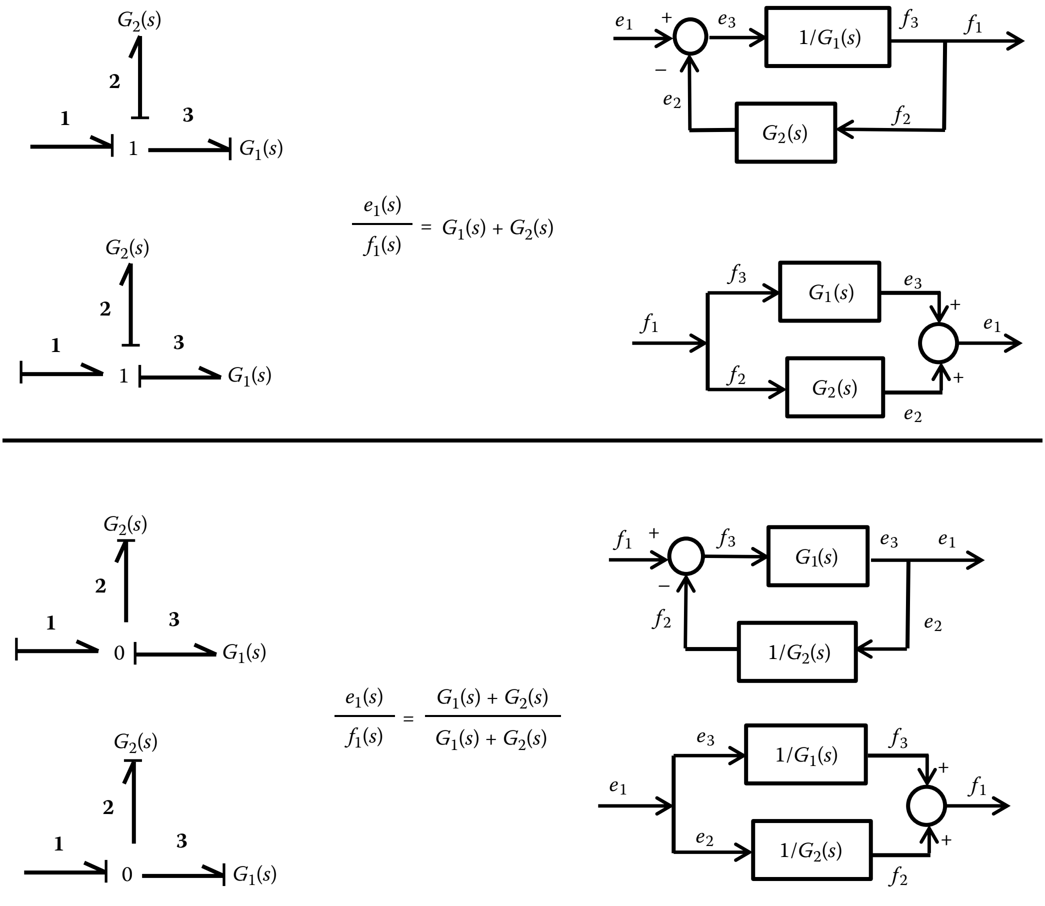

$G_2(s)$
2
1
3
1
$G_1(s)$
e_1
+
e_3
$1/G_1(s)$
f_3
f_1
−
e_2
$G_2(s)$
f_2
$\frac{e_1(s)}{f_1(s)} = G_1(s) + G_2(s)$
$G_2(s)$
2
1
3
1
$G_1(s)$
f_3
$G_1(s)$
e_3
+
f_1
e_1
f_2
$G_2(s)$
+
e_2
$G_2(s)$
2
1
3
0
$G_1(s)$
f_1
+
f_3
$G_1(s)$
e_3
e_1
−
f_2
$1/G_2(s)$
e_2
$\frac{e_1(s)}{f_1(s)} = \frac{G_1(s) + G_2(s)}{G_1(s) + G_2(s)}$
$G_2(s)$
2
1
3
0
$G_1(s)$
e_3
$1/G_1(s)$
f_3
+
e_1
f_1
e_2
$1/G_2(s)$
+
f_2

FIGURE 9.13
Bond graph model of feedback and feed-forward control structures and their block diagrams.

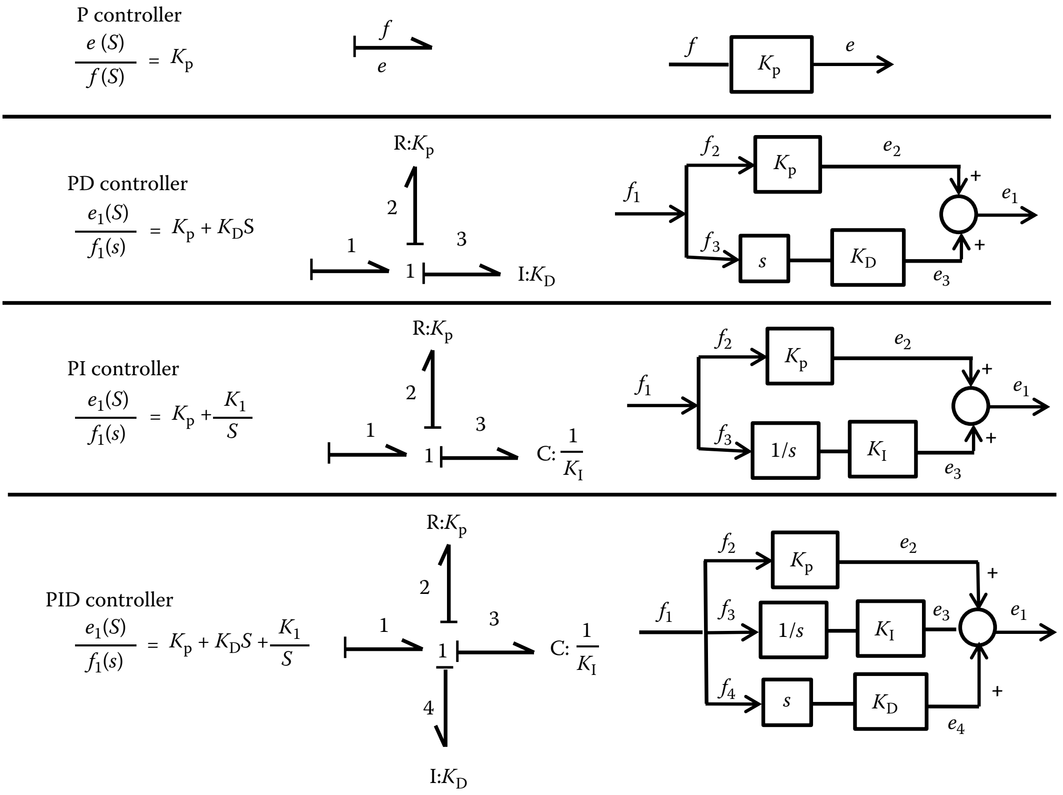

P controller
e(S)/f(S) = Kp
f
e
Kp
PD controller
e1(S)/f1(s) = Kp + KDS
R:Kp
1
2
3
I:KD
f1
f2
f3
s
KD
e1
e2
e3
+
PI controller
e1(S)/f1(s) = Kp + KI/S
C: 1/KI
1/s
KI
PID controller
e1(S)/f1(s) = Kp + KDS + KI/S
4
f4
e4

FIGURE 9.14
Bond graph representation of different control schemes.

Zero-junction and one-junction structures in bond graphs include two general One-port elements. One of them can be considered as a feedback or feed-forward controller path, and the other one can be considered as the physical plant. In the two schemes represented in Figure 9.13, the difference of power bond causality is noticeable.

The combination of the bond graph elements C, I, and R can be used to design different control structures, such as proportional (P), proportional plus derivative (PD), proportional plus integral (PI), and proportional–integral–derivative (PID), or lead and lag compensators. Therefore, a unified representation of the physical system and controller system can be prepared for the concurrent design of the mechatronic systems. The bond graph models of some impedance controller schemes are shown in Figure 9.14 with their associated transfer functions and block diagrams [14].

The application of this method is presented in the controller design problems in Sections 9.4 and 9.5.

9.4 Two-Loop Design Model

The topology realization process requires remarkable effort and creativity by experts. As mentioned in Section 9.2, an integration of bond graph modeling and genetic programming is used to make this process autonomous. Using this integrated tool, the topology realization and sizing optimization processes may be carried out simultaneously. The associated design model is shown in Figure 9.2.

In the design model shown in Figure 9.2, the topology optimization and the sizing optimization are performed simultaneously and in the same level. However, the experience of natural evolution of an engineering system indicates that the system topology evolves at a much slower rate than the parametric value improvement.

In order to clarify this point, consider the natural evolution procedure of a typical machine or mechatronic system during a period of time. In most generation renovations, only an insignificant level of evolution with minor structural modification takes place. In fact, one may observe that in most generations, the optimum design of the previous generation has been improved only slightly and insignificantly with minor revisions to the previous design. In contrast, usually after passing several numbers of generations of insignificant evolution, an evolution with significant modification in the structure of the system takes place, which is usually termed an innovation. This evolution process can be represented by a two-loop optimization algorithm, which includes one inner loop and one outer loop (Figure 9.15). The outer loop represents the topological improvements of the generation individuals, and the inner loop only shows the numerical changes, which are in fact minor and insignificant modifications. It can be stated that the repetition rate of the inner loop is quiet faster than that of the outer loop.

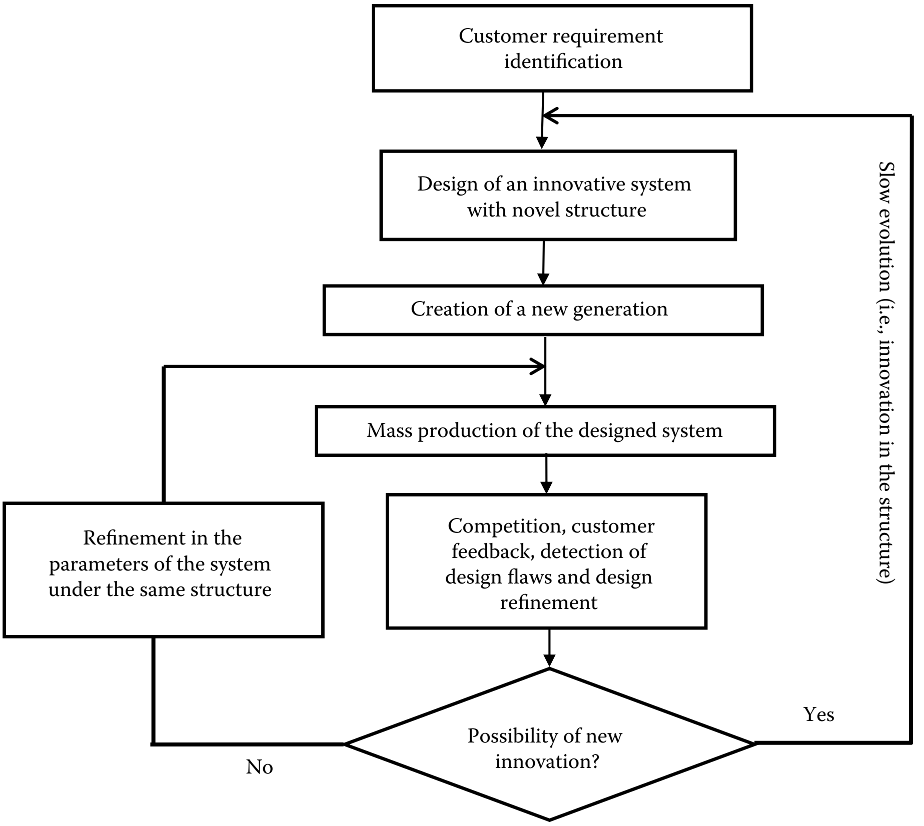

Customer requirement identification
Design of an innovative system with novel structure
Creation of a new generation
Mass production of the designed system
Refinement in the parameters of the system under the same structure
Competition, customer feedback, detection of design flaws and design refinement
Possibility of new innovation?
No
Yes
Slow evolution (i.e., innovation in the structure)

FIGURE 9.15
Flowchart of a natural evolution of an engineering system.

A comparison of the two flowcharts in Figures 9.2 and 9.15 indicates that the design model of the early integrated tool does not agree with the natural evolution process. In this model, the evolutionary search for the best topology and sizing are performed in the same level, and in the natural evolution model, the minor modifications (i.e., numerical changes) are carried out at a faster rate compared to the major modifications (i.e., topology realization).

In a genetic algorithm, the randomly generated individuals compete according to their associated fitness values, and the solutions with higher ranks will survive and are allowed to participate in the creation of the next generation. On the other hand, the topology realization is the most important and essential part of the optimization because this process is a very complex problem, which includes a vast search space. Despite these complexities, suppose that the optimum topology is found through random searching of GP. There is a high possibility that the fitness value of the optimum solution is low due to its inappropriate sizing. Besides, a solution with a nonoptimal topology may have a rather high fitness value. In this case, the optimum solution may lose out in the competition and be eliminated from the GP searching process. This will degrade the optimization process, and consequently a nonoptimal solution will be the result. To avoid this problem, the appropriate sizing should be established for each generated topology before participating in the competition at the topology level. It means that for each topological solution, an elite representative will compete for survival. Then a more reasonable decision would be made about each individual ranking and the possibility of participating in the creation of the next generation.

9.4.1 Hybrid Genetic Algorithm with Genetic Programming

In order to overcome the mentioned drawbacks of the early design model, a two-loop optimization tool has been proposed by Behbahani and de Silva in 2013 [15]. In this methodology, the bond graph modeling is integrated with a hybrid of GA and GP. This model is presented as a flowchart in Figure 9.16. It is observed that the new two-loop design model is more compliant with the natural evolution model (Figure 9.15).

The model consists of an outer loop that is associated with topology realization using GP and an inner loop, which is exerted for the optimization of parametric values. The steps of this model are as follows:

1. *N* random topologies are generated by using the integration of BG and GP tools.
2. For each generated topology, the parametric values are counted. Then, an evolutionary competition is accomplished by considering its parametric values as the optimization parameters, and the elite of that topology is represented as the result. This step corresponds to the inner loop optimization, which is performed by a simple GA. The fitness value of the elite is considered as the fitness of that topology.
3. The elites of the *N* topologies compete, and the solutions with better fitness values are given greater probability to participate in the reproductive operation for creating the new generation of topologies (outer loop).
4. Similar to step 2, the elites of each of the solutions in the new generation are found (i.e., inner loop with GA). Similar to step 3, these elites compete, and consequently a new generation is created (i.e., outer loop with GP). These steps are followed until the termination conditions are satisfied.

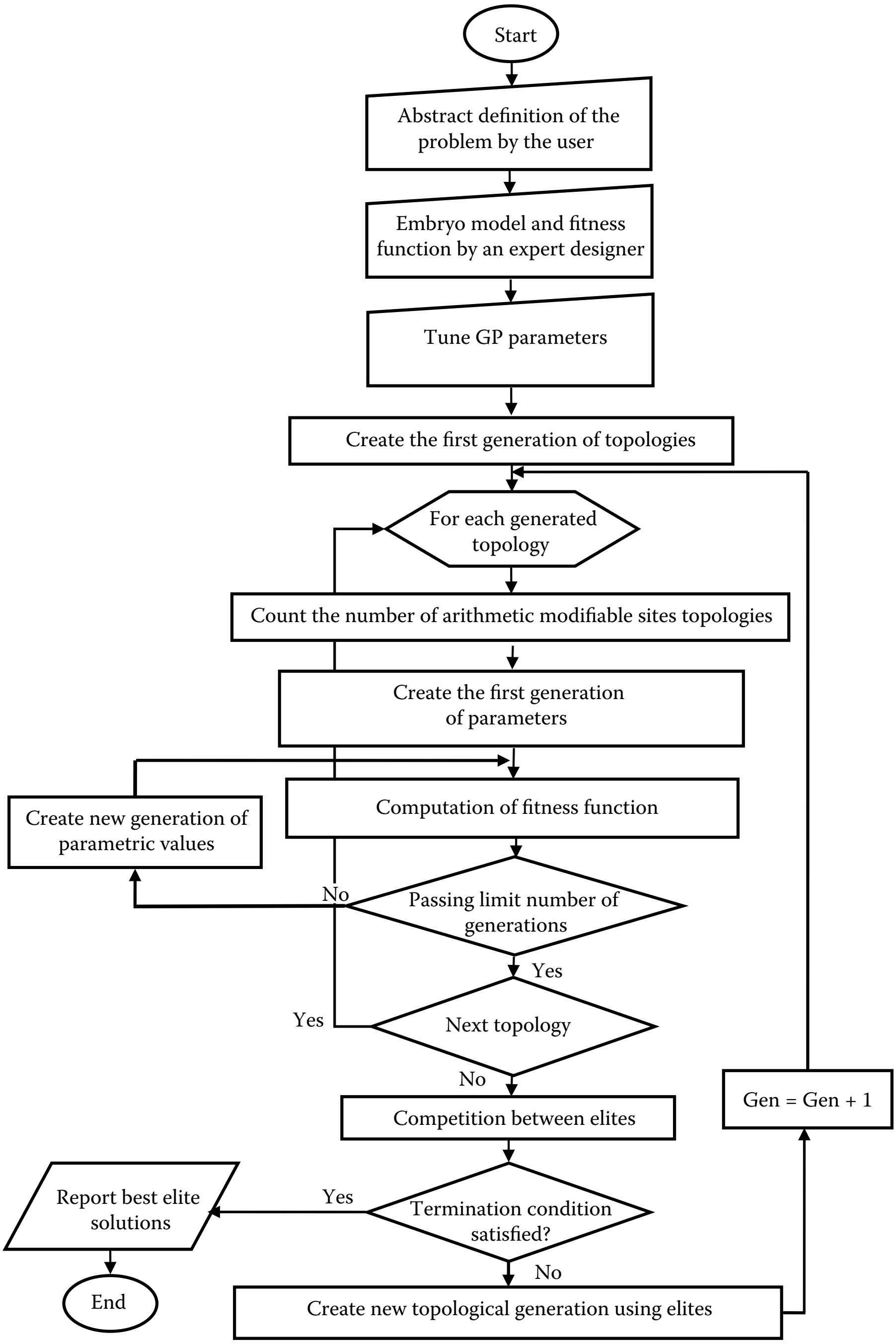

Start
Abstract definition of the problem by the user
Embryo model and fitness function by an expert designer
Tune GP parameters
Create the first generation of topologies
For each generated topology
Count the number of arithmetic modifiable sites topologies
Create the first generation of parameters
Create new generation of parametric values
Computation of fitness function
No
Passing limit number of generations
Yes
Yes
Next topology
No
Competition between elites
Gen = Gen + 1
Report best elite solutions
Yes
Termination condition satisfied?
No
End
Create new topological generation using elites

FIGURE 9.16
Two-loop design model for mechatronic system synthesis.

Unlike the earlier version of the integration of BG and GP, arithmetic sites are not used in the tree-like representation of each solution. Instead, a GA is utilized for the optimization of the new added parameters and the parameters of the embryo that are considered as modifiable elements.

9.4.2 Case Study: Iron Butcher Controller Design [15]

The iron butcher is an automated machine that has been designed and built in the Industrial Automation Laboratory of the University of British Columbia (UBC) to automatically cut the heads of fish with maximum meat recovery (Figure 9.17).

A vision system is used to find the position of the gill of a fish, and the obtained coordinates are sent to a two-axis positioning table to move the cutter to the optimum location. The actuation system in the positioning table includes two electrohydraulic manipulators operated by two servo valves. A schematic diagram of the electro-hydraulic system is shown in Figure 9.18. This electrohydraulic system is also used as the test bed for several research projects that are aimed at developing advanced control technologies to cope with such factors as nonlinear friction, system parameter variation, and communication time delay.

In this example, the objective is to automatically design an appropriate controller for the machine. In the first stage, a high-level definition of the desired behavior should be provided for this highly nonlinear system. A comprehensive model of the system is presented

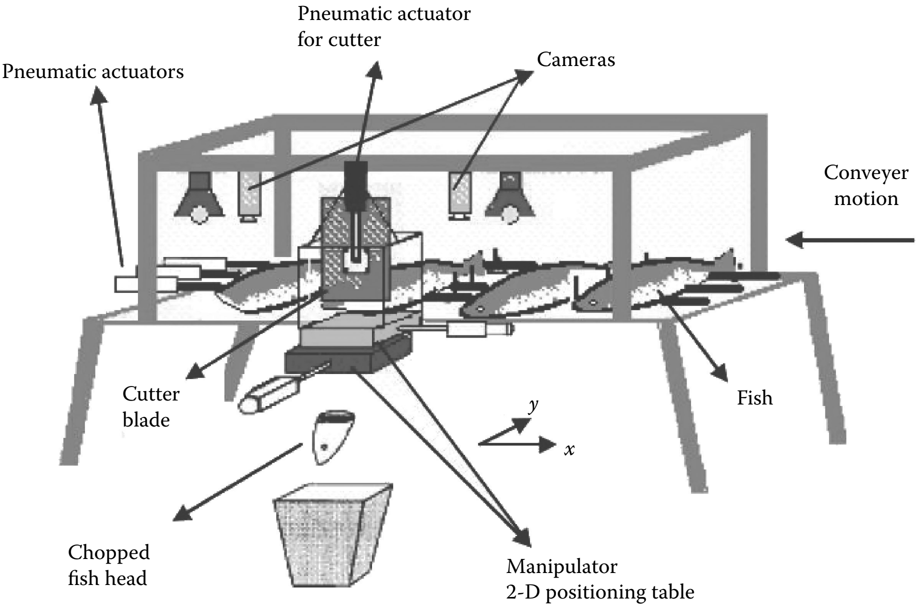


FIGURE 9.17
Schematic diagram of the iron butcher.

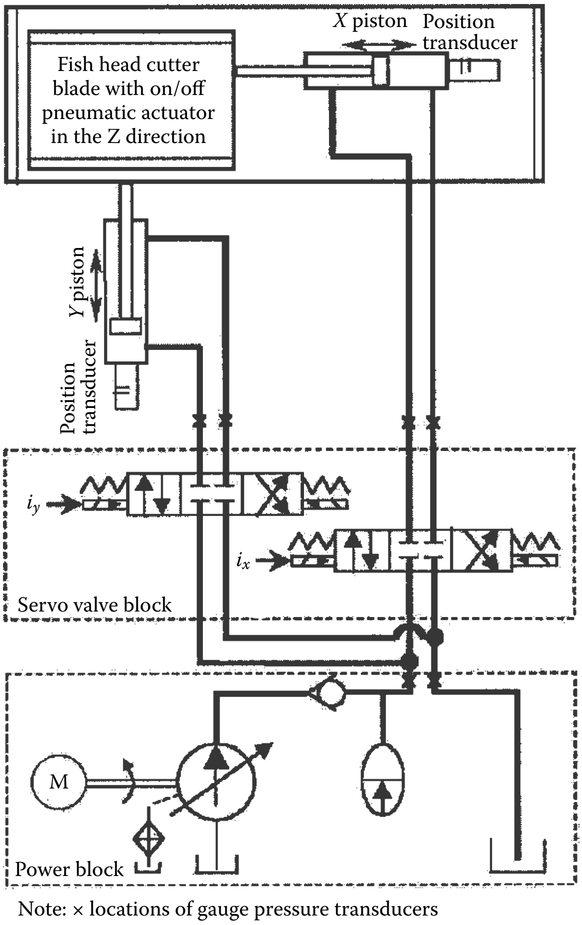

X piston
Position transducer
Fish head cutter blade with on/off pneumatic actuator in the Z direction
Y piston
Position transducer
i_y
i_x
Servo valve block
M
Power block
Note: × locations of gauge pressure transducers

FIGURE 9.18
Schematic diagram of the electrohydraulic manipulator.

in [1], and a simplified model is used in this problem because the model is simulated so many times during the optimization process.

The block diagram of the original system is shown in Figure 9.19. The current to the servo valve (i) is the input, and the position is the measured output of the system. The external force from the hydraulic system is calculated by measuring the pressure at both sides of the piston (P_1 and P_2) and multiplying them by the piston areas (A_1 and A_2). Friction (F_f) is considered as a disturbance to the system. Friction is an important challenge in controlling

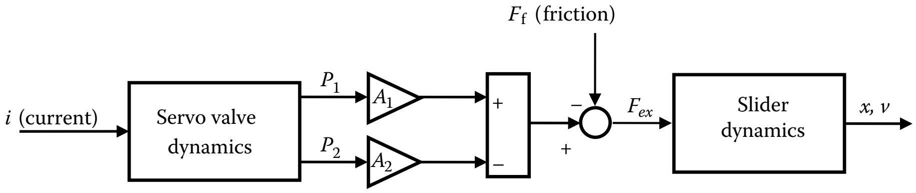

F_f (friction)
i (current)
Servo valve dynamics
P_1
P_2
A_1
A_2
+
−
−
+
F_{ex}
Slider dynamics
x, v

FIGURE 9.19
Block diagram of the original system.

the system because it highly affects the dynamic behavior of the system. It is desirable to present an estimation of the friction and velocity to the controller in each time step. Therefore, a friction observer as well as a velocity observer have been incorporated into the system.

Figure 9.20 represents the block diagram of the observers. The pole placement method is used to design the velocity observer. The convergence of the observation must be rather fast in comparison to the controller, so the poles of the observer are placed at −200 and −300 to guarantee their quick convergence.

In this example, a single-causality model is utilized to observe the friction from the velocity, which is capable of accounting for the essential and complex characteristics of the friction, including hysteresis, stick-slip, presliding displacement, frictional lag, and varying breakaway force. Microscopic bristles between contacting surfaces can be considered to justify the mathematical model. The stick and presliding behavior caused by friction is due to the deflection of the bristles in the form of small cantilever beams, which will resist the relative motion of the object. After the force passes a specific threshold value (i.e., breakaway force), the bristles deflect sufficiently to allow the object to begin sliding. The mathematical formulations of the friction model are given below [16–18]:

$$\begin{gathered}
\frac{\mathrm{d}z}{\mathrm{d}t} = v - \frac{|v|}{g(v)} z \\
g(v) = \frac{F_\mathrm{c} + (F_\mathrm{s} - F_\mathrm{c})e - \left(\dfrac{v}{v_0}\right)^2}{\sigma_0} \\
F_\mathrm{f} = \sigma_0 z + \sigma_1 \frac{\mathrm{d}z}{\mathrm{d}t} + \sigma_2 v
\end{gathered} \tag{9.3}$$

Here v is the relative velocity, F_f is the frictional force, F_c is the Coulomb friction, F_s is the stick friction force, v_s is the Stribeck velocity, σ_0 and σ_1 are stiffness and damping coefficients of the bristles, σ_2 represents viscous friction, and z is an internal variable of the model that represents the average bristle deflection.

The block diagram of the controller is shown in Figure 9.21. At each time step, the measured position, observed velocity (outputs for feedback control), and friction (disturbance

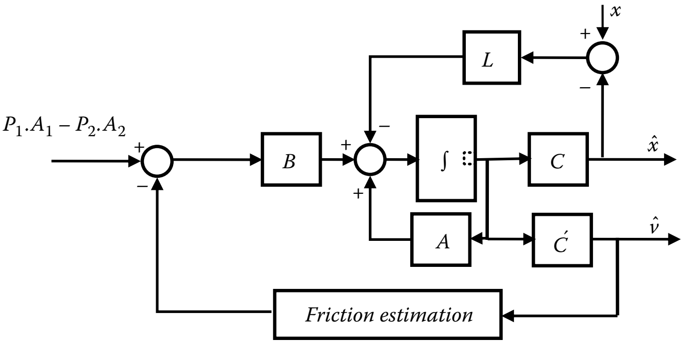


FIGURE 9.20
Block diagram of the observers.

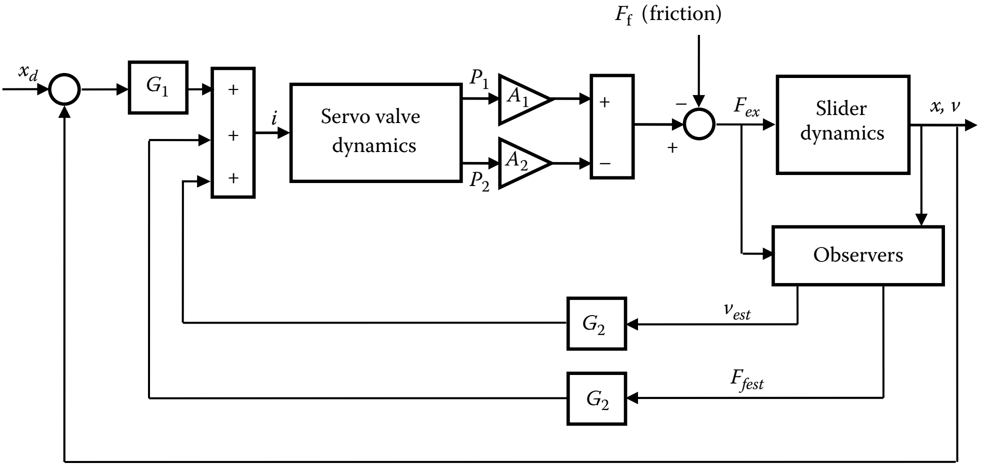


FIGURE 9.21
Block diagram of the control system.

for feed-forward control) are available for the controller. In order to design the controller, the transfer functions G_1, G_2, and G_3 have to be synthesized.

In this stage, a reference model is used to evaluate the performance of the controller. In fact, the objective is to develop a controller with a performance as close as possible to the reference model. A first-order system with the following transfer function is taken as the reference model:

$$G_{\text{Ref}} = \frac{1}{0.02s+1} \tag{9.4}$$

Using the evolutionary mechatronic tool, the transfer functions G_1, G_2, and G_3 were synthesized. The evolved controller has indicated that the compensation of the friction is the best way to control the system. Specifically the G_3 controller, which is multiplied by the estimated friction, contributes to the current so that the force generated by current eliminates the friction. For the system with the designed controller, Figure 9.22 shows the

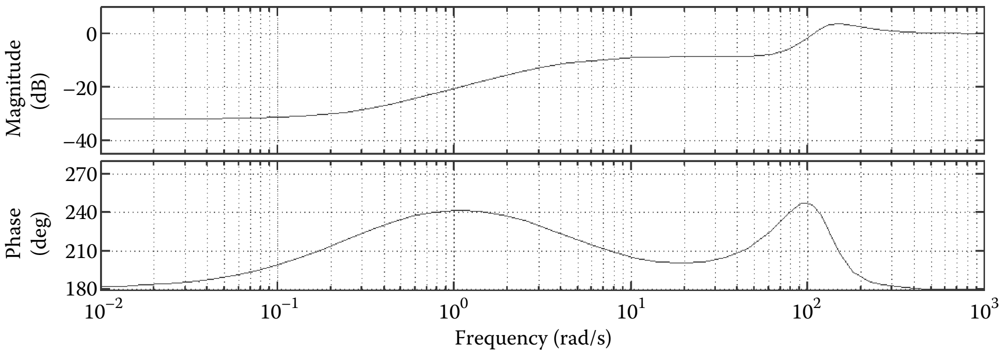


FIGURE 9.22
Bode diagram between friction and output position.

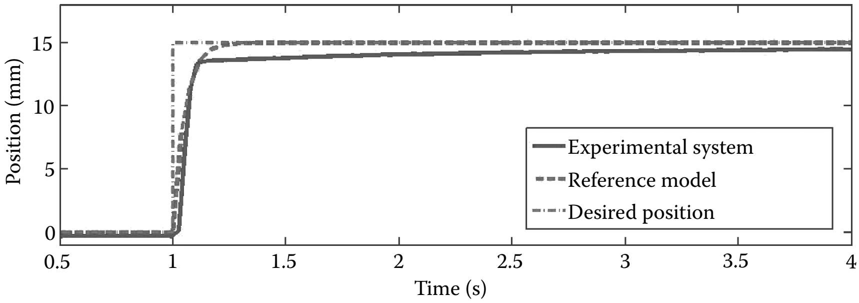


FIGURE 9.23
Performance of the designed controller.

Bode diagram of the transfer function between the friction and the output position. The Bode diagram demonstrates that the automatically synthesized controller can reduce the effect of friction on the output. This concludes that the controller compensates the friction successfully.

A comparison of the designed controller performance and the experimental performance of the controlled system with the reference model performance is presented in Figure 9.23. It can be seen that the desired performance is closely achieved by the automatically designed controller.

9.5 Niching Optimization Scheme

The evaluation of individuals in the outlined design methodology is based on a fitness value, which is assumed to be programmable to be computed automatically in the optimization loop of the procedure. However, in a practical engineering design problem, it is not always possible to define a proper programmable fitness function, which includes all the essential designing criteria. There may be some subjective criteria involved in a design problem, and for the evaluation of some design criteria, such as cost, reliability, and life expectancy, we may need to refer to manufacturer's catalogues. Many such criteria cannot be evaluated automatically in the optimization loop of the program.

In addition, because different consumers have different preferences about the criteria that are involved in a design problem, the desirable aggregation method of these criteria would not be distinctive. Therefore, it is more favorable for consumers to choose from different types and brands of any product in the market based on their own preferences and limitations.

It can be claimed that in a practical design of a mechatronic structure, several optimal solutions with considerable different topologies must be presented to the designer, so that the designer can evaluate them according to the nonprogrammable criteria and make the final decision. Then, products with different structures, costs, and performance can be presented so that the consumers will have more comprehensive and desirable choices with respect to their own preferences and limitations.

Niching genetic algorithm is a concept in biologically inspired genetic algorithms, which can handle multimodal optimization. In this method, the individuals with some similarities are grouped as colonies, and in order to find the elites of these colonies, a competition is performed among them. Clearly, this concept matches the requirement of providing several alternative solutions for a practical mechatronic design.

9.5.1 Niching Genetic Programming

In 2014, Behbahani and de Silva [19] developed a methodology for multimodal topology optimization. This algorithm can be categorized as a niching genetic algorithm and is based on the strategy of restricted competition selection (RCS) [20–22]. The two-loop structure of their previous work [15], which was a combination of BG, GP, and GA, is utilized to explore the complex search space of mechatronic design. In an RCS niching genetic algorithm, the individuals who belong to different colonies do not compete with each other if they are considerably different from each other. However, there is an evolutionary competition between the individuals inside a colony, which eventually forces the individuals to go toward the elite of the colony. Assuming that the fitness function is defined so that the objective is to find the solutions with maximum fitness, the steps of this algorithm are as follows:

1. N random topologies are generated by using the tool that integrates BG and GP.
2. The elite of each individual topology is explored considering its parametric values. A simple GA is used in this step.
3. The fitness of each elite is considered to be the fitness of its associated topology, and M fittest topologies are selected through a competition. An elite set is formed, which is transferred to the next topology competing set.
4. N new random topologies are created by applying the common reproductive operations on the prior generation.
5. The new generation and the elite set that is established in step 3, create a competing set with $N + M$ topologies.
6. Similar to step 2, a simple GA is used to explore the representative of the topology of each individual solution of the previously formed competing set. The fitness of each elite is considered as the fitness of its associated topology.
7. This is the main step of executing the RCS strategy. Each trial topology is compared with all other generated topologies, and the level of their similarity is evaluated. If their structural difference is less than a predefined threshold, a competition is performed between them and the fitness of the loser is assigned as zero. Eventually, the fitness of most of the individuals will be zero.
8. The new elite set is generated by the topologies with nonzero fitness.
9. In order to create the next generation, to be the parents, N solutions are selected randomly from the competing set. It is desirable to include the elite set among the new parents.
10. The terminating condition is checked, and if it is not satisfied, the process is continued by going to step 4; otherwise, the process is stopped, and the elite set is presented to the designer.

In this algorithm, all the generated topologies have an equal chance for participating in the parent set. It can be stated that, when a trial topology is considerably different from other topologies, its elite is allowed to be represented in the elite set.

9.5.2 Case Study: Model-Referenced Active Car Suspension [19]

In this section, the automatic synthesis of an active car suspension is performed as an illustrative example. The main function of the car suspension system is to provide passenger comfort by minimizing the transmitted vertical acceleration. In fact, road handling, load carrying, and passenger comfort are three conflicting criteria that are involved in the car suspension design. A quarter car model shown in Figure 9.24 is commonly used for the analysis and design of a car suspension system due to its simplicity and ability to provide some important characteristics of the full model.

Generally, in a quarter car suspension model, the main vehicle body mass is represented by the sprung mass, and the tires and axels are represented as the unsprung masses. In Figure 9.24, M_s and M_u are sprung mass and unsprung mass, respectively. The tire is considered as a spring with stiffness K_t. The actuator force is denoted by u. The load and the road disturbances are the two main sources of disturbance acting on the car, which are denoted by X_s and X_r, respectively. Corresponding equations of motion for the quarter car suspension system are as follows:

$$\begin{aligned} M_s\ddot{X}_s &= -u + F_L \\ M_u\ddot{X}_u &= u + K_t(X_r - X_u) \end{aligned} \tag{9.5}$$

Then the matrix G can be obtained from Equation 9.5 and the specified suspension force u:

$$\begin{bmatrix} F_r \\ \dot{X}_s \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} \dot{X}_r \\ F_L \end{bmatrix} \tag{9.6}$$

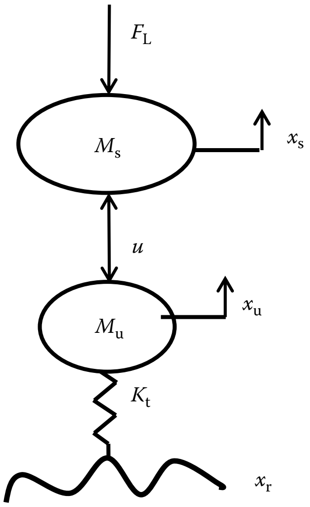


FIGURE 9.24
Schematic diagram of quarter car suspension model.

Here F_r denotes the force that is transmitted between the tire and the road. In this problem, it is assumed that both road and load disturbances exist. Then, the velocity of the sprung mass and the relative oscillation between the sprung and unsprung masses are measured and sent to the controller as the feedbacks of the system. The controller law is taken as

$$u = K_{\mathrm{I}}(s)(x_s - x_u) + K_{\mathrm{II}}(s)\dot{x}_s \tag{9.7}$$

Here, $K_{\mathrm{I}}(s)$ and $K_{\mathrm{II}}(s)$ are two unknown controllers. Their structures and inherent parameters have to be synthesized.

In this design example, a reference model is used in order to provide the desired and effective dynamic characteristics of the system. Skyhook damping is a common and well-accepted concept for car suspension design [23–25]. The principle of this method is to consider a theoretical situation in which the sprung mass is linked with a damper to an imaginary point in the sky. Because a stationary point for fixing the damper is not practically available, the designed system is not practically implementable in a passive form. Even in active implementation, the dynamic specifications of the unsprung mass will deteriorate because the control force is applied to both sprung and unsprung masses.

For this design problem, the dynamic features computed through the double skyhook approach are considered to be the reference model. The objective is to design an appropriate controller for implementation in a passive form, resulting in a system response that is as close as possible to the response of the reference model. The double skyhook configuration may be stated as

$$u = K_s(X_s - X_u) + C_1\dot{X}_s - C_2\dot{X}_u \tag{9.8}$$

Here, C_1 and C_2 correspond to the skyhook dampers for the sprung and unsprung masses, respectively, and K_s is the stiffness of a spring between the sprung mass and the unsprung mass. The suspension system has to be soft with regard to the road disturbances, and it has to be hard against the load disturbances. Considering these characteristics, the desired transfer function between the rider movement and the road $\left(H_1(s) = \dfrac{\dot{X}_s}{\dot{X}_r}\right)$ and the load disturbance $\left(H_2(s) = \dfrac{\dot{X}_s}{F_L}\right)$ can be obtained by choosing suitable parameters in the presented configuration of the following equations:

$$\begin{aligned} H_1(s) &= \frac{c_2 k_t \cdot s + k_s \cdot k_t}{m_s \cdot m_u s^4 + (c_1 \cdot m_u + c_2 \cdot m_s)s^3 + (k_s m_u + k_s m_s + k_t m_s)s^2 + c_1 k_t \cdot s + k_s k_t} \\ H_2(s) &= \frac{(m_u s^2 + c_2' \cdot s + k_s' + k_t)s}{m_s \cdot m_u s^4 + (c_1' \cdot m_u + c_2' \cdot m_s)s^3 + (k_s' m_u + k_s' m_s + k_t m_s)s^2 + c_1' k_t \cdot s + k_s' k_t} \end{aligned} \tag{9.9}$$

The parameters for computation of the desired reference model response are considered similar to what is reported in [13]. The parameters of the quarter car model are taken as $M_s = 250$ Kg, $M_u = 35$ Kg, and $K_t = 150 \times 10^3$ N/m, and the parameters for the skyhook configuration are presented in Table 9.2.

TABLE 9.2

The Double Skyhook Configuration Parameters

Disturbance	K_s (N/m)	C_1 (Ns/m)	C_2 (Ns/m)
Road	1000	4000	2000
Load	150,000	12,000	6000

TABLE 9.3

Evolved Elite Controllers Developed by Niching Genetic Programming

Elite Number	Controller Transfer Function
Elite #1	$K_{\mathrm{I}}(s) = \dfrac{1806s^2 + 6.684 \times 10^4 s + 4.009 \times 10^4}{s}$ $K_{\mathrm{II}}(s) = \dfrac{9501s + 8.463 \times 10^4}{s}$
Elite #2	$K_{\mathrm{I}}(s) = 2117s + 6.875 \times 10^4$ $K_{\mathrm{II}}(s) = \dfrac{6.67 \times 10^4 s^4 + 6.57 \times 10^{16} s^3 + 8.86 \times 10^{17} s^2 + 2.68 \times 10^{18} s + 3.7 \times 10^{17}}{s^4 + 7.86 \times 10^{12} s^3 + 5.03 \times 10^{13} s^2 + 3.04 \times 10^{13} s}$
Elite #3	$K_{\mathrm{I}}(s) = 2168s + 6.024 \times 10^4$ $K_{\mathrm{II}}(s) = \dfrac{1.033 \times 10^5 s^3 + 1.305 \times 10^{11} s^2 + 1.344 \times 10^{12} s + 1.27 \times 10^{10}}{s^3 + 2.16 \times 10^7 s^2 + 1.334 \times 10^7 s + 1.23 \times 10^5}$

The two controller transfer functions $K_{\mathrm{I}}(s)$ and $K_{\mathrm{II}}(s)$ are automatically designed. In the calculation of the fitness function of each generated model, a point-by-point comparison is made between the Bode diagrams of the developed controllers and the response of the reference model. In this process, both magnitude and phase value are compared for the two transfer functions. Three resulting elite sets of the controllers are presented in Table 9.3.

Figure 9.25 presents Bode diagrams of the system with the automatically designed controllers, compared with the reference model. It is observed that the desired performance that is achieved by the double skyhook approach, which is hardly implementable, can be obtained by the developed controllers, which have implementable structures. The users can choose the most desirable and appropriate configuration by considering their own preferences and limitations.

9.5.3 Case Study: Hydraulic Engine Mount Design

Engine mounts are used to suppress the transmission of vibration disturbances between two structures, which are connected by an engine mount as an interface [26]. Another function of engine mounts is to support the weight of the engine. Figure 9.26 shows a schematic for a typical single pumper hydraulic engine mount. It consists of two rubber chambers on the top and the bottom, and they are connected through a fluid channel, called the inertia track. Depending on the viscosity of the fluid, two different objectives are pursued. A high-viscosity fluid may be utilized in some applications to create additional damping, and in some systems a low-viscosity fluid is used to provide a tuned vibration absorber effect [27]. In this example, the second objective is considered.

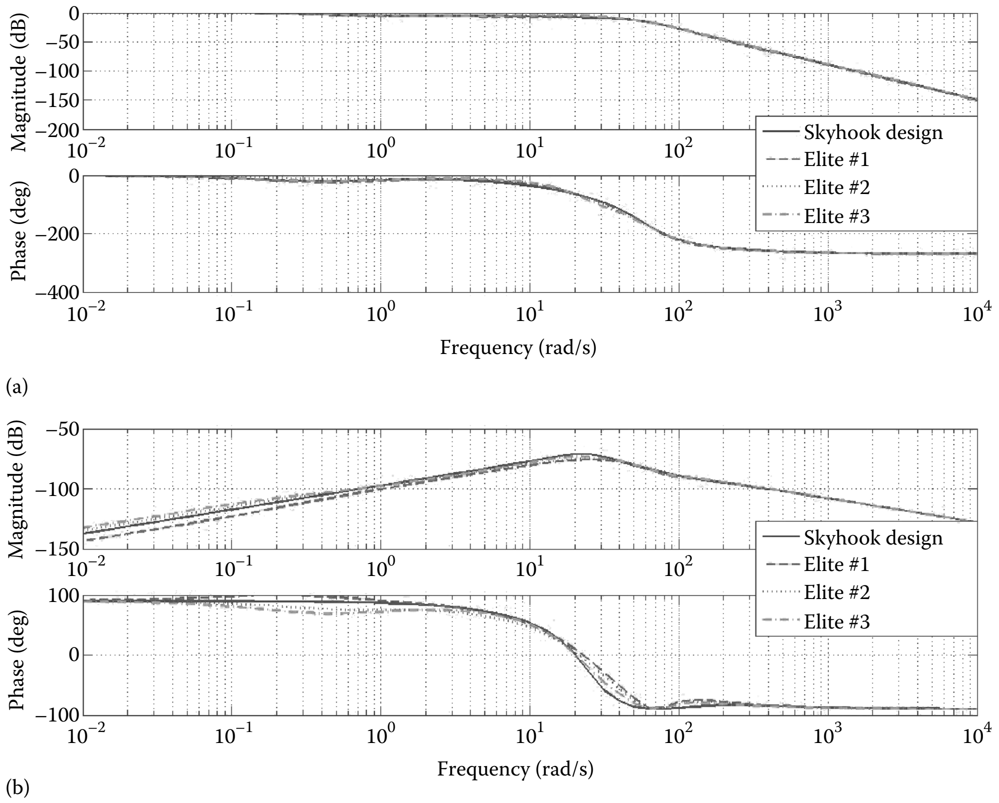


FIGURE 9.25
Comparison of the frequency response of evolved controllers with the ideal reference model: (a) $H_1(s)$ and (b) $H_2(s)$.

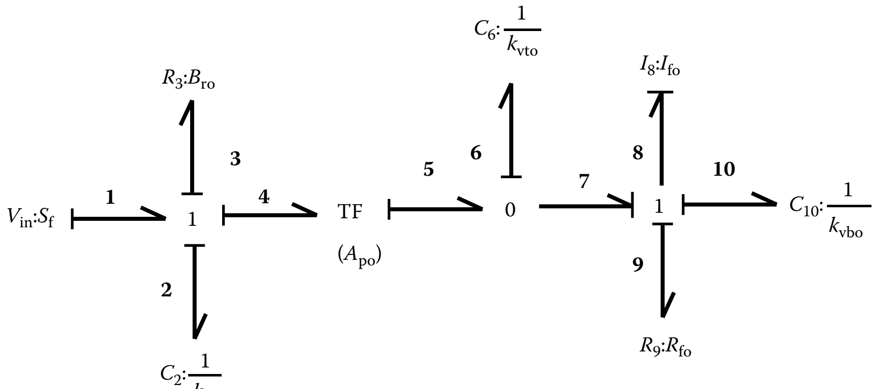

V_{in}:S_f
1
1
2
C_2:$\frac{1}{k_{ro}}$
3
R_3:B_{ro}
4
TF
(A_{po})
5
0
6
C_6:$\frac{1}{k_{vto}}$
7
1
8
I_8:I_{fo}
9
R_9:R_{fo}
10
C_{10}:$\frac{1}{k_{vbo}}$

FIGURE 9.26
Bond graph model of a typical single pumper engine mount.

The presence of a peak frequency is a defect of these systems, which corresponds to the post-resonance of the fluid moving along the inertia track (see Figure 9.27). But, for engine mounts, there exists a frequency of very low dynamic stiffness, which is called the notch frequency. It is desirable to tune the notch frequency to the disturbance frequency, to reject disturbances. Obviously, a deeper notch will lead to better vibration isolation.

In the optimization process of engine mount systems, two main objectives are considered. The first objective is to lower the dynamic stiffness at the peak frequency in order to reduce disturbance transmission at high frequencies. The second objective is to generate a deeper notch. However, a contradiction exists in simultaneously achieving these objectives. Specifically, lowering the peak will make the notch shallower, and conversely, reducing the dynamic stiffness at notch frequency will generate a higher peak [28]. Therefore, maintaining a balance between these two objectives is considered to be an essential challenge in the optimal design of engine mounts. In this example, niching genetic programming methodology is used to synthesize the optimal topology as well as sizing for a hydraulic engine mount.

Two important requirements in this example are to consider a proper embryo model and to define an appropriate fitness function. The bond graph model of the considered embryo model is presented in Figure 9.28, which is inspired by the recently designed double-notch hydraulic engine mount [27] (see Figure 9.29).

In order to define the fitness function, the desired performance of the system should be specified. As mentioned, decreasing the dynamic stiffness at both notch and peak frequencies or reducing one of them and preventing an undesirable increase in the other one is the main objective of this design problem. However, the notch frequency should be kept unchanged to coincide with the resonant frequency, and the peak frequency should be kept far from the notch frequency. The following fitness function is determined by considering all these aspects:

$$F = \alpha_1 \frac{D_n}{D_{nt}} + \alpha_2 \frac{D_p}{D_{pt}} + P_1 + P_2 \tag{9.10}$$

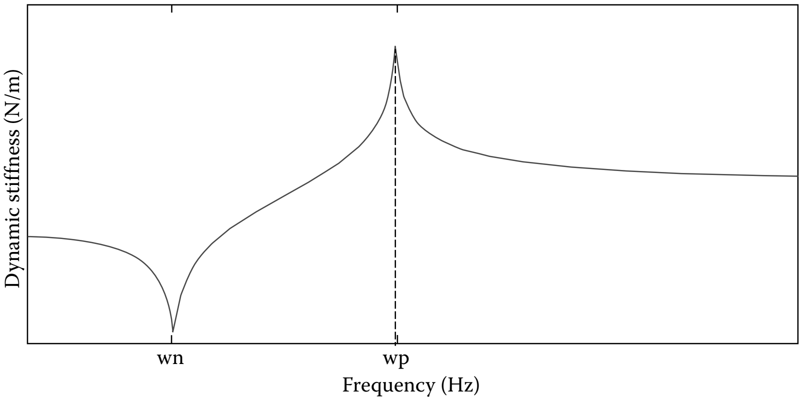


FIGURE 9.27
Dynamic stiffness of a typical engine mount.

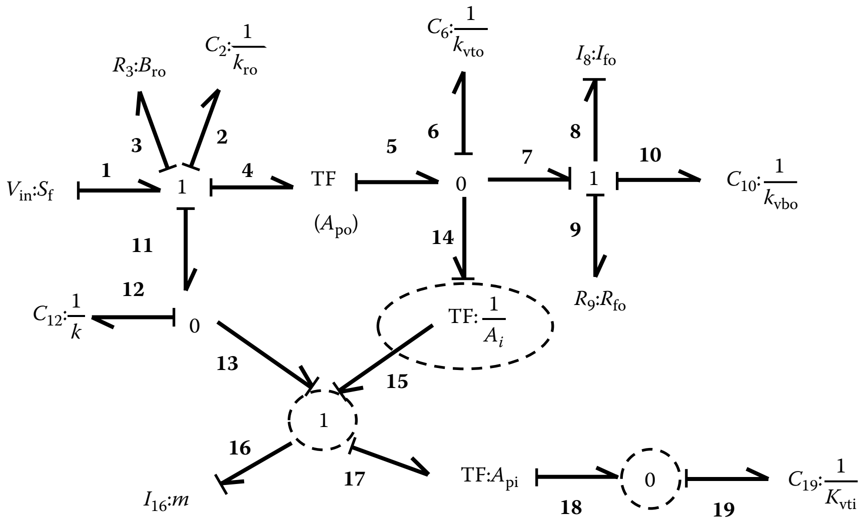

$R_3{:}B_{\text{ro}}$
$C_2{:}\frac{1}{k_{\text{ro}}}$
$C_6{:}\frac{1}{k_{\text{vto}}}$
$I_8{:}I_{\text{fo}}$
$V_{\text{in}}{:}S_{\text{f}}$
1
3
2
4
TF
(A_{po})
5
6
0
7
8
10
$C_{10}{:}\frac{1}{k_{\text{vbo}}}$
9
14
11
12
$C_{12}{:}\frac{1}{k}$
0
13
TF: $\frac{1}{A_i}$
$R_9{:}R_{\text{fo}}$
15
1
16
17
$I_{16}{:}m$
TF:A_{pi}
18
0
19
$C_{19}{:}\frac{1}{K_{\text{vti}}}$

FIGURE 9.28
Bond graph model of the embryo.

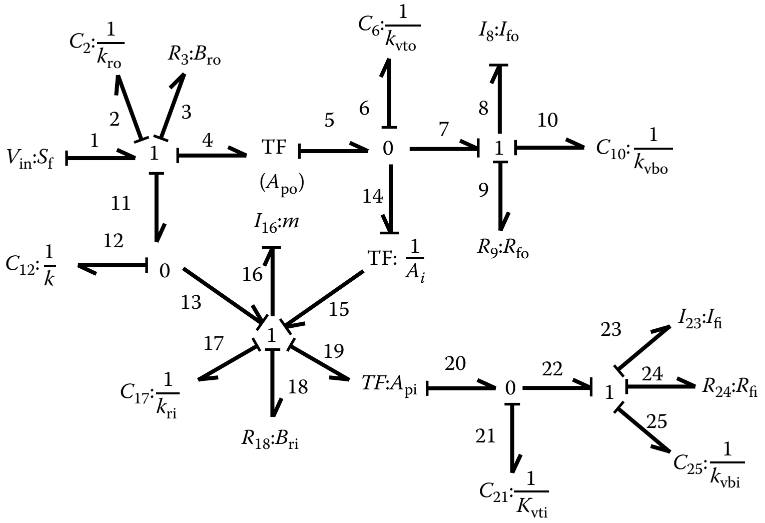

$C_2{:}\frac{1}{k_{\text{ro}}}$
$R_3{:}B_{\text{ro}}$
$C_6{:}\frac{1}{k_{\text{vto}}}$
$I_8{:}I_{\text{fo}}$
$V_{\text{in}}{:}S_{\text{f}}$
1
2
3
4
TF
(A_{po})
5
6
0
7
8
1
10
$C_{10}{:}\frac{1}{k_{\text{vbo}}}$
11
14
9
$I_{16}{:}m$
12
$C_{12}{:}\frac{1}{k}$
0
16
TF: $\frac{1}{A_i}$
$R_9{:}R_{\text{fo}}$
13
15
17
1
19
18
$C_{17}{:}\frac{1}{k_{\text{ri}}}$
$R_{18}{:}B_{\text{ri}}$
TF:A_{pi}
20
0
22
1
23
$I_{23}{:}I_{\text{fi}}$
24
$R_{24}{:}R_{\text{fi}}$
25
$C_{25}{:}\frac{1}{k_{\text{vbi}}}$
21
$C_{21}{:}\frac{1}{K_{\text{vti}}}$

FIGURE 9.29
Bond graph model of double-notch hydraulic engine mount. (From Tikani, R., Vahdati, N., Ziaei-Rad, S., and Esfahanian, M., *Journal of Vibration and Control*, 17, 11, 1644–1656, 2011.)

Here, D_n and D_p are the values of the dynamic stiffness at the notch and peak frequencies, respectively, and equivalent amounts of a typical engine mount of similar parameter values are used to normalize them (D_{nt} and D_{pt}); P_1 and P_2 represent penalty coefficients, which are used to satisfy the required conditions for notch and peak frequencies; and α_1 and α_2 are the weighting coefficients.

Figure 9.30 shows the bond graph models of the elite solutions generated by the evolutionary tool.

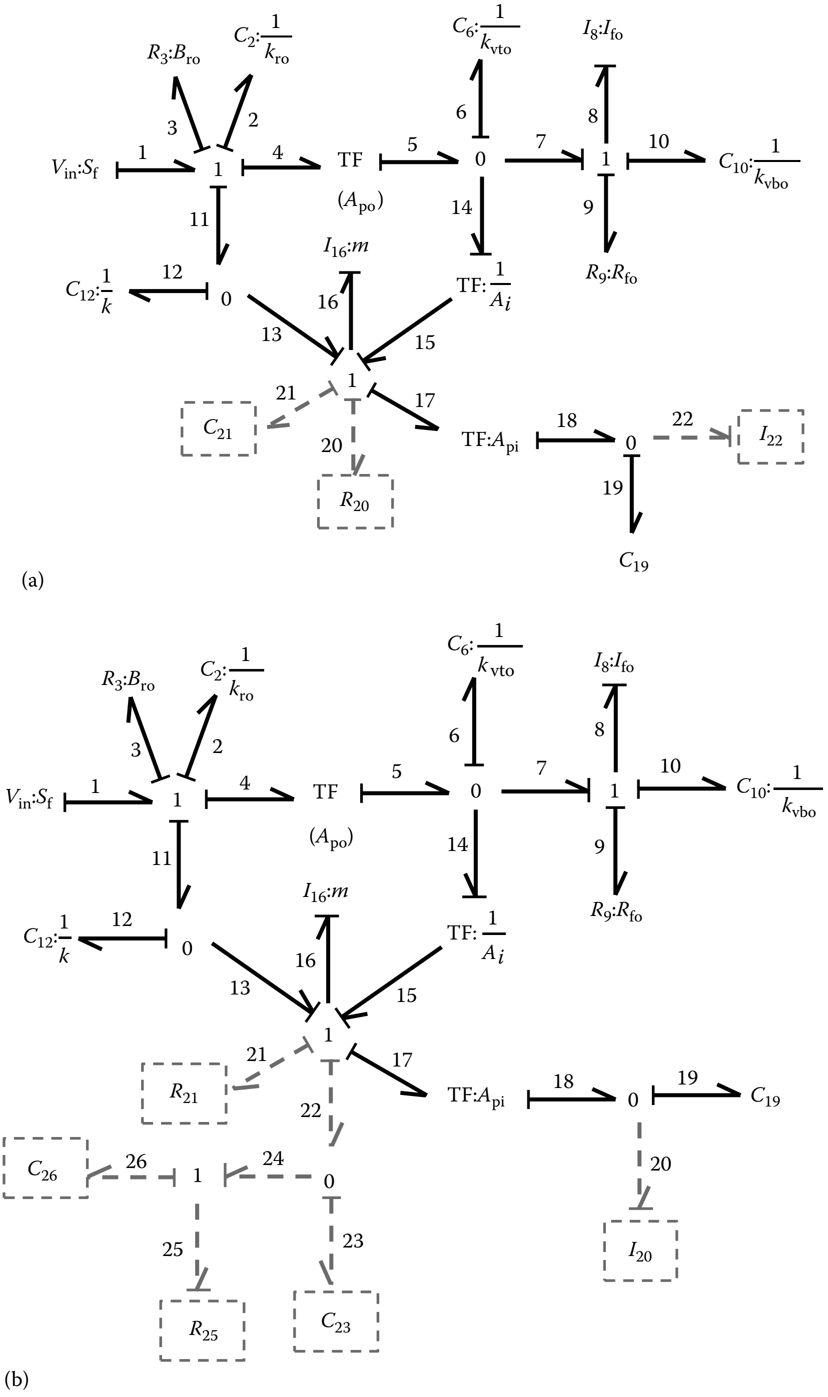

(a)
V_{in}:S_f
R_3:B_{ro}
C_2:$\frac{1}{k_{ro}}$
C_6:$\frac{1}{k_{vto}}$
I_8:I_{fo}
TF
(A_{po})
C_{10}:$\frac{1}{k_{vbo}}$
R_9:R_{fo}
C_{12}:$\frac{1}{k}$
I_{16}:m
TF:$\frac{1}{A_i}$
TF:A_{pi}
C_{21}
R_{20}
I_{22}
C_{19}
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
(b)
V_{in}:S_f
R_3:B_{ro}
C_2:$\frac{1}{k_{ro}}$
C_6:$\frac{1}{k_{vto}}$
I_8:I_{fo}
TF
(A_{po})
C_{10}:$\frac{1}{k_{vbo}}$
R_9:R_{fo}
C_{12}:$\frac{1}{k}$
I_{16}:m
TF:$\frac{1}{A_i}$
TF:A_{pi}
R_{21}
C_{19}
C_{26}
I_{20}
R_{25}
C_{23}
23
24
25
26

FIGURE 9.30
Bond graph models of the elite solutions.

(*Continued*)

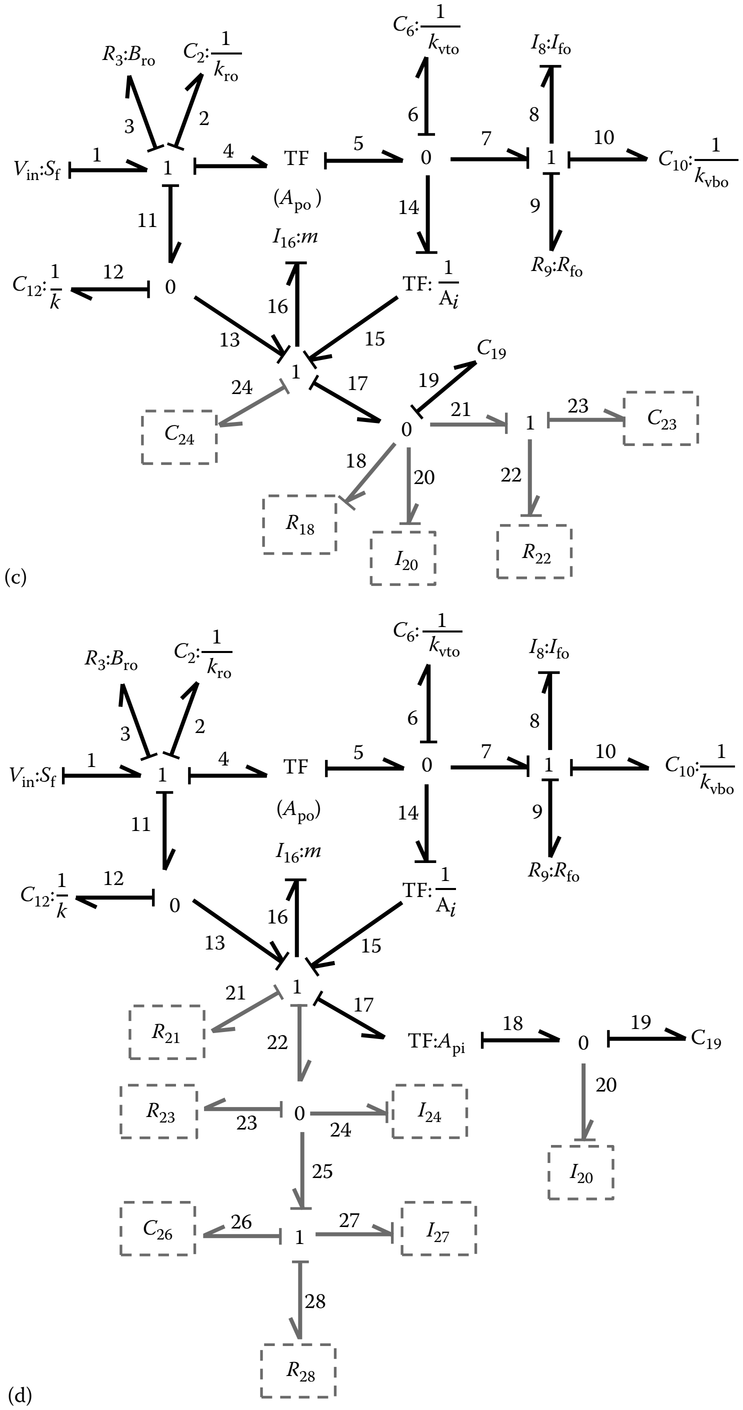

V_{in}:S_f
R_3:B_{ro}
C_2:$\frac{1}{k_{ro}}$
C_6:$\frac{1}{k_{vto}}$
I_8:I_{fo}
C_{10}:$\frac{1}{k_{vbo}}$
R_9:R_{fo}
TF
(A_{po})
I_{16}:m
TF: $\frac{1}{A_i}$
C_{12}:$\frac{1}{k}$
C_{19}
C_{24}
C_{23}
R_{18}
I_{20}
R_{22}
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
0
1
(c)
V_{in}:S_f
R_3:B_{ro}
C_2:$\frac{1}{k_{ro}}$
C_6:$\frac{1}{k_{vto}}$
I_8:I_{fo}
C_{10}:$\frac{1}{k_{vbo}}$
R_9:R_{fo}
TF
(A_{po})
I_{16}:m
TF: $\frac{1}{A_i}$
C_{12}:$\frac{1}{k}$
TF:A_{pi}
C_{19}
R_{21}
R_{23}
I_{24}
I_{20}
C_{26}
I_{27}
R_{28}
25
26
27
28
(d)

FIGURE 9.30 (CONTINUED)
Bond graph models of the elite solutions.

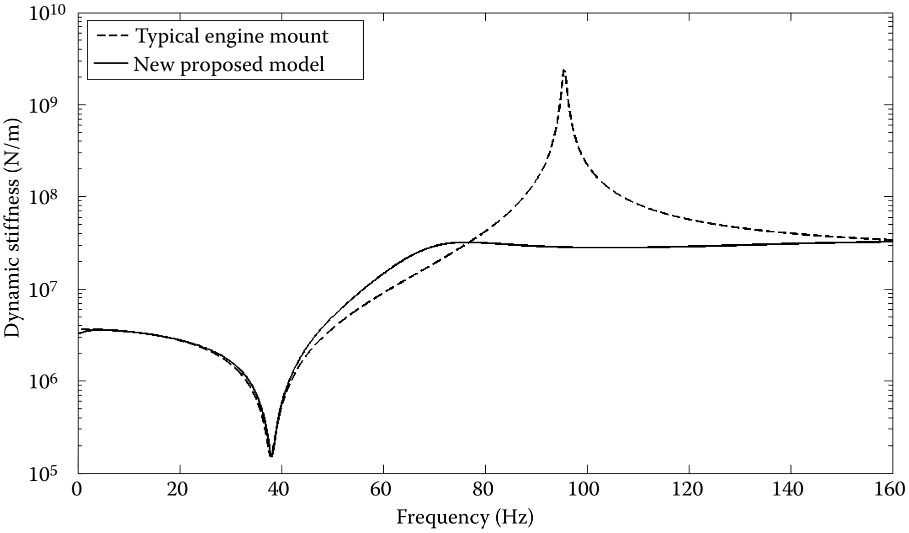


FIGURE 9.31
Comparison of the dynamic stiffness of the new engine mount designs and a typical engine mount.

A comparison of the dynamic stiffness of the elite solutions and a typical hydraulic engine mount is presented in Figure 9.31 (due to the similarity of the diagrams just one of them is shown).

It is seen that the resonant peak is eliminated while the notch frequency and the dynamic stiffness in this frequency are preserved without any undesirable changes. The similarity of the performances is another important characteristic of the evolved models. It means that four considerably different structures of hydraulic engine mounts with similar acceptable performances are suggested to the designer by the evolutionary tool. Different physical models can be represented by these bond graph models, according to the designer's limitations and preferences.

Tikani et al. [27] presented two different performances for their hydraulic engine mount design by changing the working fluid characteristics. In the first design, they used a common fluid, which resulted in two notch frequencies and two peak frequencies. In the second design, they utilized a tunable high viscosity fluid (MR* or ER† fluid), which resulted in the elimination of the peak frequency. Figure 9.32 shows a comparison of the dynamic stiffness of the elite solutions and the two different performances of the hydraulic engine mount proposed in [27]. By examining the notch and peak frequencies and their dynamic stiffness, it is clear that the resulted topologies of this design example offer considerably better performance.

---

[*] Magneto-rheological.

[†] Electro-rheological.

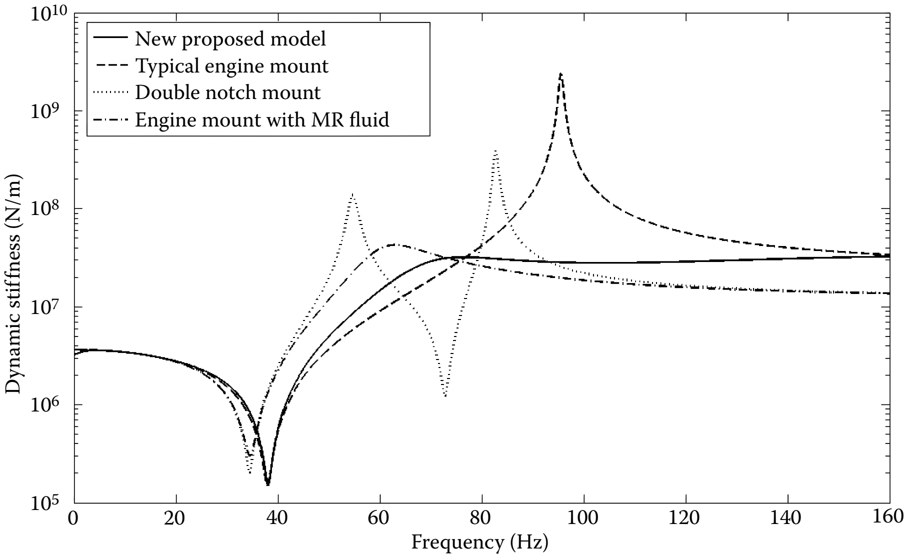


FIGURE 9.32
Comparison of the dynamic stiffness of new engine mount design with double-notch design and design with MR fluid of [27].

9.6 Conclusions

Design and modeling of mechatronic systems are challenging tasks due to the presence of complex subsystems and interacting components in different domains. Therefore, for such systems, the designer must employ concurrent and integrated approaches. This chapter presented an integration of bond graphs and genetic programming resulting in a multipurpose evolutionary tool for the design and identification of multidomain systems in an optimal manner. This optimization tool was extended in order to develop a suitable methodology for the automatic synthesis of mechatronic systems. A two-loop structure was proposed by linking GP and GA to imitate the natural evolution of systems. In the outer loop, GP was utilized for topology realization, and in the inner loop, the elite solution of each topology was found by using GA for optimizing the parametric values. As a more advanced approach, a multi-modal niching genetic algorithm was developed for the competition between topologies. The tool represented several optimal solutions with different topologies, in compliance with the presence of nonprogrammable criteria in topology evaluation. The proposed improvements were inspired by the natural evolution processes. The obtained results in the presented examples were quite encouraging, which supported extension of the developed tool to provide a more comprehensive mechatronic design tool.

References

1. de Silva, C. W., *Mechatronics—A Foundation Course*, CRC Press, Taylor & Francis, Boca Raton, FL, 2010.

2. Koza, J. R., Bennett, F. H., Andre, D., and Keane, M. A., "Synthesis of topology and sizing of analog electrical circuits by means of genetic programming," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2, pp. 459–482, 2000.
3. Koza, J. R., Bennett, F. H., Andre, D., and Keane, M. A., *Genetic Programming III, Darwinian Invention and Problem Solving*, Morgan Kaufman Publication, San Francisco, 1999.
4. Seo, K., Fan, Z. N., Hu, J., Goodman, E. D., and Rosenberg, R. C., "Toward a unified and automated design methodology for multi domain dynamic systems using bond graphs and genetic programming," *Mechatronics*, Vol. 13, Nos. 8–9, pp. 851–885, 2003.
5. Behbahani, S. and de Silva, C. W., "Identification of a mechatronic model using an integrated bond-graph and genetic-programming approach," *Proceedings of International Symposium on Collaborative Research in Applied Science (ISOCRIAS)*, Vancouver, Canada, pp. 158–165, 2005.
6. Granda, J. J., "The role of bond graph modeling and simulation in mechatronics systems," *Mechatronics*, Vol. 12, Nos. 9–10, pp. 1271–1295, 2002.
7. Amerongen, J. V., "Mechatronic design," *Mechatronics*, Vol. 13, No. 10, pp. 1045–1066, 2003.
8. Karnopp, D., Margolis, D. L., and Rosenberg, R. C., *System Dynamics: Modeling and Simulation of Mechatronic Systems*, Wiley, New York, 2000.
9. GP tutorial. Available at: http://www.geneticprogramming.com/Tutorial/.
10. Behbahani, S., *Practical and Analytical Studies on Development of Formal Evaluation and Design Methodologies for Mechatronic Systems*, PhD Thesis, Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada, 2006.
11. Introduction to genetic algorithms. Available at: http://cgm.cs.mcgill.ca/~soss/cs644/projects/marko/introduction.html.
12. Sharon, A., Hogan, N., and Hardt, D. E., "Controller design in the physical domain," *J. The Franklin Inst.*, Vol. 328, No. 5/6, pp. 697–721, 1991.
13. Wang, J., Fan, Z., Terpenny, J. P., and Goodman, E. D., "Knowledge interaction with genetic programming in mechatronic systems design using bond graphs," *IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 35, No. 2, pp. 172–182, 2005.
14. Wang, J., *Integrated Coevolutionary Synthesis of Mechatronic Systems Using Bond Graphs*, PhD Thesis, Industrial Engineering and Operations Research, The University of Massachusetts, Amherst, MA, 2004.
15. Behbahani, S. and de Silva, C. W., "Mechatronic design evolution using bond graphs and hybrid genetic algorithm with genetic programming," *IEEE/ASME Transactions on Mechatronics*, Vol. 18, No. 1, pp. 190–199, 2013.
16. Canudas de Wit, C., Olsson, H., Astrom, K. J., and Lischinsky, P., "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, Vol. 40, No. 3, pp. 419–425, 1995.
17. Lee, T. H., Tan, K. K., and Huang, S., "Adaptive friction compensation with a dynamical friction model," *IEEE/ASME Transactions on Mechatronics*, Vol. 16, No. 1, pp. 133–140, 2011.
18. Berger, E. J., "Friction modeling for dynamic system simulation," *Applied Mechanics Review*, Vol. 55, No. 6, pp. 535–577, 2002.
19. Behbahani, S. and de Silva, C. W., "Niching genetic scheme with bond graphs for topology and parameter optimization of a mechatronic system," *IEEE/ASME Transactions on Mechatronics*, Vol. 19, No. 1, pp. 269–277, 2014.
20. Behbahani, S. and de Silva, C. W., "A new multi-criteria mechatronic design methodology using niching genetic algorithm," *Proceedings of IEEE World Congress on Evolutionary Computation*, Vancouver, Canada, pp. 1031–1036, 2006.
21. Kim, J. K., Cho, D. H., Jung, H. K., and Lee, C. G., "Niching genetic algorithm adopting restricted competition selection combined with pattern search method," *IEEE Transaction on Magnetics*, Vol. 38, No. 2, pp. 1001–1004, 2002.
22. Cho, D. H., Jung, H. K., and Lee, C. G., "Induction motor design for electric vehicle using a niching genetic algorithm," *IEEE Transaction on Industry Applications*, Vol. 37, No. 4, pp. 994–999, 2001.

23. Parthasarathy, S. S. and Srinivasa, Y. G., "Design of an active suspension system for a quarter-car road vehicle model using model reference control," *Proceedings of the Institution of Mechanical Engineers (IMechE), Part I: J. Systems and Control Engineering*, Vol. 220, pp. 91–108, 2006.
24. Pacchionia, A., Goodallb, R. M., and Brunia, S., "Active suspension for a two-axle railway vehicle," *Vehicle System Dynamics*, Vol. 48, pp. 105–120, 2010.
25. Ziyaeifar, M., "Vibration control in train-bridge-track systems," *Vehicle System Dynamics*, Vol. 46, No. 5, pp. 355–384, 2008.
26. De Silva, C. W., *Vibration—Fundamentals and Practice*, 2nd Edition, Taylor & Francis, CRC Press, Boca Raton, FL, 2007.
27. Tikani, R., Vahdati, N., Ziaei-Rad, S., and Esfahanian, M., "A new hydraulic engine mount design without the peak frequency," *Journal of Vibration and Control*, Vol. 17, No. 11, pp. 1644–1656, 2011.
28. Ahn, Y. K., Kim, Y. C., Yang, B. S., Ahmadian, M., Ahn, K. K., and Morishitah, S., "Optimal design of an engine mount using an enhanced genetic algorithm with simplex method," *Vehicle System Dynamics*, Vol. 43, No. 1, pp. 57–81, 2005.

10

Design Evolution of Mechatronic Systems

Lalith B. Gamage

CONTENTS

10.1 Introduction 360
10.2 Modeling Multidomain Systems 361
    10.2.1 Bond Graph Modeling 361
        10.2.1.1 Causality Assignment 363
        10.2.1.2 State-Space Formulation 365
    10.2.2 Linear Graphs 368
10.3 Design Evolution 371
    10.3.1 Evolutionary Design Framework with BGs 372
    10.3.2 Methodology 374
    10.3.3 Solution Representation for the Evolution 374
    10.3.4 Fitness Function 375
10.4 Application of Methodology to Industrial Systems 377
    10.4.1 Illustrative Scenario 1 382
        10.4.1.1 Fitness Function 383
        10.4.1.2 System Identification 384
    10.4.2 Illustrative Example of Application of LG Methodology 386
        10.4.2.1 Modeling and Simulation of the Conveying System 387
    10.4.3 Illustrative Scenario 2 395
        10.4.3.1 LG Evolution with GP 395
        10.4.3.2 Design Improvement of the Conveying System 398
10.5 Conclusions 401
References 401

SUMMARY This chapter presents a system framework for design evolution of multidomain engineering systems or mechatronic systems. The described framework integrates machine health monitoring with an expert system to monitor the performance of an existing mechatronic system and to make a decision as to whether a design improvement is necessary. If a design improvement is necessary, the expert system will also assist in identifying the section of the system that needs to be improved. Modeling of multidomain engineering systems using bond graphs and linear graphs is explained in detail with examples. A technique to evolve a system model represented by a bond graph or a linear graph using genetic programming is introduced. This technique is combined with a fitness function that represents a set of desired performance specifications to obtain an optimum design. The developed approach allows exploration of the design space to find the optimum design solution in an automated manner. In order to control the arbitrary exploration of the design space, domain knowledge, expertise, and input from the

machine health monitoring system are used. The design evolution algorithm is implemented using GPLAB, a MATLAB® tool, and integrated with Simscape™ for modeling and simulation. The developed system is applied to an industrial fish-processing machine to demonstrate its capability in addressing performance issues by redesigning two subsystems of the machine. Also, to illustrate the usability of the two modeling techniques, bond graph modeling and linear graph modeling were employed to redesign the hydraulic cutter positioning system and electromechanical conveyor system of the fish-processing machine, respectively.

10.1 Introduction

A mechatronic system is a mixed and multidomain engineering system, which would consist of components or parts belonging to domains, such as electrical, mechanical, and hydraulic [1]. Approaches to modeling, design, and control of a mixed system should somewhat deviate from those for conventional, nonmechatronic systems in part due to the presence of interactions between components and the need to integrate several types of energy transfer [2].

The design of a multidomain mechatronic system should take an integrated and concurrent approach rather than a standalone approach in which subsystems in each domain are designed separately, neglecting the dynamic interactions among them. The traditional approach has been to design domain-specific subsystems, such as the electrical subsystem and mechanical subsystem separately and to integrate them to form the design for the entire system. However, a design formulated this way would not be optimal as important interactions among domains may have been neglected. Therefore, design and optimization should be carried out taking the entire system into consideration [3].

In a typical mechatronic design optimization problem, the objective is to find the optimal design of a system, which should include the simplest possible structure (or topology) of the system and the corresponding parameter values of its components such that the final design best satisfies the user requirements. The topology is defined as the gross number of components, their type, and the way they are interconnected in the system while parameter values are the numerical values of the elements. For example, if the user requirement were to design a low-pass filter with a certain cutoff frequency, the design objective would be to design a system with the simplest possible topology with the least number of components and to determine their values such that the cutoff frequency requirement is met. As the topology of practical mechatronic systems is not usually apparent, the design optimization problem can be complex. Evolutionary methodologies can be employed to explore the design space, which consists of a large number of design alternatives, to find the optimum design solution in an automated manner. As evolutionary methodologies can be open-ended, which can lead to arbitrary exploration of design space, experience and expertise can be used to restrict the design space only to plausible regions. A design engineer's experience and expertise can be captured into a knowledge-based system that can be used to intelligently control the evolution process. Also, the knowledge-based system can enhance its knowledge by monitoring the behavior of the system over time. A machine health monitoring system can be employed to monitor the performance of the machine, and performance data from this system can be analyzed to enhance the knowledge base [4]. A schematic representation of the evolutionary framework is shown in Figure 10.1.

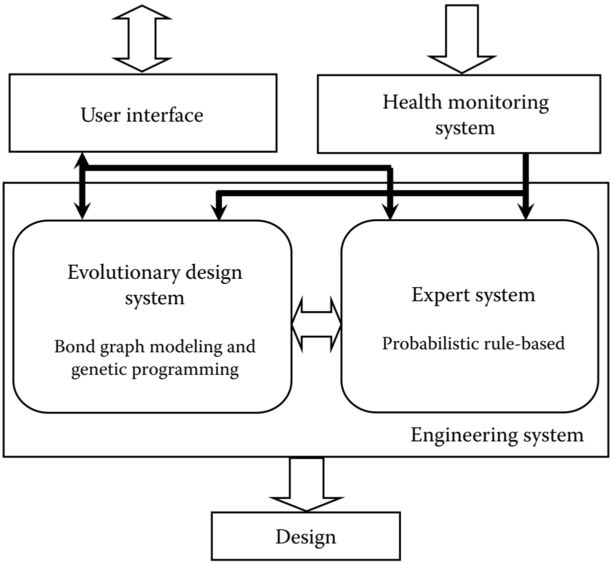


FIGURE 10.1
Schematic diagram of the design evolution system.

This approach was first proposed by de Silva in 2008 [5] and subsequently further developed by Gamage, de Silva, Campos, and others [6,7].

In order to apply evolution methodologies, the multidomain mechatronic system must be defined or modeled using a common language that can represent components in different domains. Put another way, electrical, mechanical, hydraulic, and other components must be modeled using a unified methodology such that the entire system can be evolved as a whole to find an optimum solution.

10.2 Modeling Multidomain Systems

10.2.1 Bond Graph Modeling

Bond graph (BG) modeling is a multidomain modeling technique that is useful in modeling and simulation of a multidomain engineering or mechatronic system [3,8]. Bond graphs are graphical representations of lumped parameter models of dynamic engineering systems, which provide a unified approach to modeling and analysis of such systems. BGs are made of elements and bonds. There are four types of elements: one-port passive elements, one-port active elements, two-port elements, and junctions. One-port passive elements are C, I, and R elements, which, in the electrical domain, represent capacitors, inductors, and resistors, and, in the mechanical domain, represent springs, masses, and dampers, respectively. One-port active elements are effort and flow sources of power S_e and S_f, which, in the electrical domain, represent voltage and current and, in the mechanical domain, force and velocity, respectively. There are two two-port elements, transformers and gyrators, represented by TF and GY, respectively. The fourth type of element are "0" and "1" junctions with three or more ports, which are used to interconnect other elements using bonds.

There are two variables known as effort and flow variables in a bond graph. The effort and flow variables in different energy domains are shown in Table 10.1. Regardless of the domain, the product of these two variables is equal to power. BG models can describe the dynamic behavior of systems based on the principles of conservation of power.

A line segment with a half arrow, which connects two elements in a BG model, represents a bond. The half arrow indicates the direction of power flow. For sources, the power flow is always from the source toward the system. Therefore, the half arrow is directed outward from the source element toward the system. Because a passive element consumes energy, the half arrow is directed toward the element. The vertical line segment that accompanies the arrow in a bond is called the "causality stroke," which is used to indicate the input and output variables of an element. If the causality stroke is at the near end of the element, the input to that element is the effort variable, and the output of the element is the flow variable. If the causality stroke is at the far end of the element, the input is the flow variable, and the output is the effort variable. For example, if element A in Figure 10.2 represents an element in an electrical circuit, voltage (effort) will be the input to this element and current (flow) will be the output.

A junction that connects other elements in a BG indicates which variable (effort or flow) is common to the elements attached to the junction and is used to represent continuity and compatibility relationships. The net power flow into a junction is equal to zero. In the case of a "1" junction, the flow variables of the elements attached to the junction are identical, and the effort variables add up to zero, representing a continuity or conservation relationship. The effort variables of the elements attached to a "0" junction are the same, and the summation of flow variables is zero, and this represents a compatibility condition.

Consider the spring, mass, and damper system shown in Figure 10.3 to illustrate the construction of a BG. The force F (effort source S_e) is acting on mass m_1 (inertia element L_1), leading to the first common velocity ("1") junction. As mentioned, the velocity around this junction is the same. The effort through the spring element k_1 (capacitive element C_1) is the same, creating the next force ("0") junction. The velocities of mass m_2 (inertia L_2), spring k_2 (capacitive element C_2), and viscous damper b_2 (resistive element R_2) are the same. They are attached to the last velocity ("1") junction, completing the BG.

Similarly, a BG can be established for the electrical circuit shown in Figure 10.3 as well. The voltage V (effort source S_e) causes a current through inductor L_1, leading to the first common current ("1") junction. The voltage across capacitor C_1 and the series connected

TABLE 10.1

Effort and Flow Variables in Different Energy Domains

Domain	Effort	Flow
Mechanical	Force	Velocity
Electrical	Voltage	Current
Fluid	Pressure	Flow rate

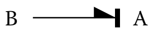


FIGURE 10.2
A representation of a bond between elements A and B with causality stroke.

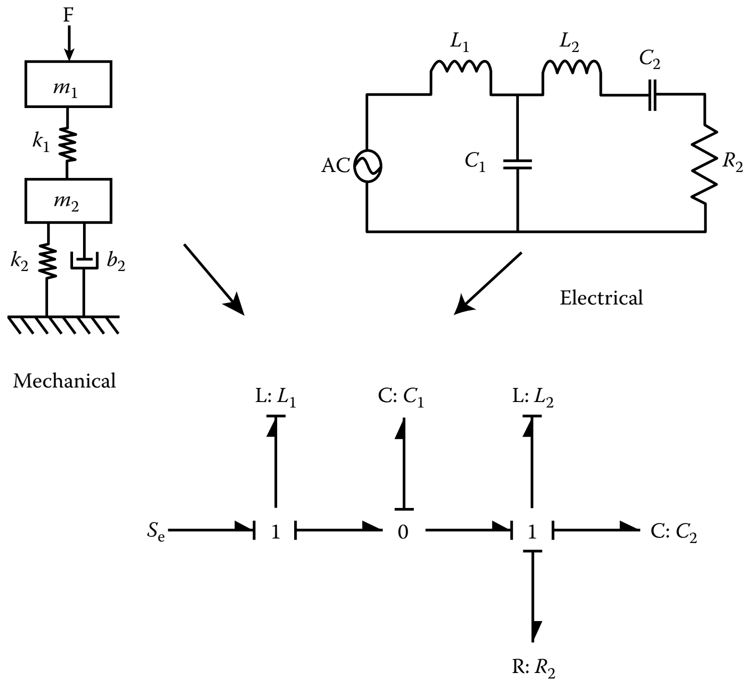


FIGURE 10.3
BG model of an electrical and a mechanical system.

L_2, C_2, and R_2 section is the same, creating a common voltage ("0") junction. As the current through the series connected L_2, C_2, and R_2 is the same, it creates the last current ("1") junction of the BG.

The interdisciplinary nature of BGs can be clearly seen from the above example in which two systems in electrical and mechanical domains are modeled using a single BG.

10.2.1.1 Causality Assignment

There are two forms of causality called *integral causality* and *derivative causality*. In integral causality, the integral form of the constitutive equation of an element is used [3]. For example, for a capacitive element, a constitutive equation in integral form will be as given as follows, where e and f are the effort and flow variables:

$$e = \frac{1}{c}\int f\,\mathrm{d}t$$

The derivative form of this can be given as follows:

$$f = c\frac{\mathrm{d}e}{\mathrm{d}t}$$

Often, integral causality is preferred in BG modeling due to the convenience in obtaining the state-space models and ordinary differential equations. In constructing the BG shown in Figure 10.3, we have used the integral causality. The BG representation of common elements used in BG modeling with their associated causalities is given Table 10.2.

For a common flow junction, because all the flow variables associated with that junction are the same, only one variable should be the input. Similarly, in a common effort junction, because all the efforts are the same, only one effort variable should be the input. With these rules, possible causalities for junctions are shown in Figure 10.4.

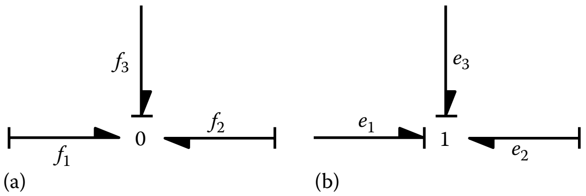
The order of a dynamic system can be determined by the number of the energy storage elements. In particular, the order of a dynamic system is equal to the number of energy storage elements. However, if there is a causality conflict, it indicates that the system order is less than the number of energy storage elements and is given by [3].

System Order = Number of Energy Storage Elements − Number of Causality Conflicts

TABLE 10.2

BG Representation of Different Elements

Mechanical	**Electrical**	**BG Representation**
Force source	Voltage source	$S_e \rightharpoonup\!\mid$
Velocity source	Current source	$S_f \mid\!\rightharpoonup$
Spring	Capacitor	$\mid\!\rightharpoonup \mathrm{C}$
Inertia	Inductor	$\rightharpoonup\!\mid \mathrm{I}$
Damper	Resistor	$\mid\!\rightharpoonup \mathrm{R}$ $\rightharpoonup\!\mid \mathrm{R}$
Mechanical transformer	Electrical transformer	$\rightharpoonup\!\mid \overset{n}{\overrightarrow{\mathrm{TF}}} \rightharpoonup\!\mid$ $\mid\!\rightharpoonup \overset{n}{\overrightarrow{\mathrm{TF}}} \mid\!\rightharpoonup$
Mechanical gyrator	Electrical gyrator	$\mid\!\rightharpoonup \overset{n}{\mathrm{GY}} \mid\!\rightharpoonup$ $\rightharpoonup\!\mid \overset{n}{\mathrm{GY}} \rightharpoonup\!\mid$



(a) (b)

FIGURE 10.4
Causalities in a junction. (a) Common effort junction and (b) common flow junction.

10.2.1.2 State-Space Formulation

The state-space model of a BG needs to be obtained in order to understand the behavior of a dynamic system. The procedure for obtaining the state-space model from a BG once the causalities are determined is given below [3].

1. Identify the independent energy-storing elements.

   In the electrical domain, the energy-storing elements are capacitors and inductors, and in the mechanical domain, they are springs and masses. If an energy-storing element is attached to a source, its state is defined by the input, and therefore, they cannot be considered as independent energy-storing elements. For example, connecting a voltage source in parallel with a capacitor (attaching a force source to a spring in series) or connecting a current source in series with an inductor (attaching a velocity source to a mass) leads to dependent states.
2. Assign state variables to independent energy-storing elements. The state variables assigned to energy-storing elements are shown in Table 10.3.
3. Select auxiliary variables that are required to formulate the state-space shell.

   For each junction in the BG, its common variable is taken as the auxiliary variable if it is not a state variable. For energy-storing elements, the variable other than its state variable will be chosen such that the total auxiliary variables can be minimized.
4. Write constitutive relationships for all elements.
5. Write continuity equations for all common flow junctions.
6. Write compatibility equations for all common effort junctions.
7. Eliminate auxiliary variables to obtain the state-space formulation.

Example 10.1

Consider the electrical system shown in Figure 10.5a. The BG model for this system is shown in Figure 10.5b. The vector **X** given below contains the variables selected in this manner for the system shown in Figure 10.5. S_e is the input, which is an effort source, and v_1, i_1, and i_2 are auxiliary variables. State variables in this system are v_c and i_L.

$$\mathbf{X} = [\dot{v}_c \; \dot{i}_L \; v_c \; i_L \; S_e \; v_1 \; i_1 \; i_2]$$

After selecting these variables, the next step would be to develop constitutive, continuity, and compatibility equations. The continuity equation for a common flow junction is obtained by equating the sum of all the effort variables to zero, and the compatibility condition for a common effort junction is satisfied by equating the sum of all the flow variables to zero. These represent the Kirchhoff's current and voltage laws in the electrical domain or in the mechanical domain geometric compatibility and dynamic

TABLE 10.3

State Variables

Capacitor	Voltage	Spring	Force
Inductor	Current	Mass	Velocity

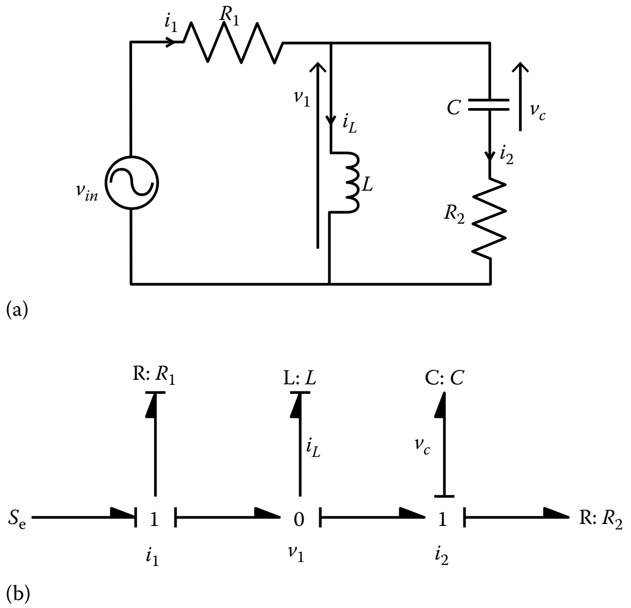


FIGURE 10.5
Electrical system and BG. (a) System and (b) bond graph model.

equilibrium of forces. These equations for the circuit in Figure 10.5a can be written as follows:

For the first common flow junction, the continuity equation is

$$S_e = i_1 R_1 + v_1$$

There will be no constitutive equations for this junction as there are no energy-storing elements attached to this junction.

For the second common effort junction, the compatibility equation is

$$i_1 = i_L + i_2$$

The constitutive equation is

$$v_1 = L i_L$$

For the last common flow junction, the continuity equation is

$$v_1 = i_2 R_2 + v_c$$

The constitutive equation is

$$i_2 = C \dot{v}_c$$

After completing these equations, one can observe that the Kirchhoff's current and voltage laws for the circuit in Figure 10.5a have been automatically taken into account. In order to eliminate the auxiliary variables, the above equations are formed into a system of linear equations as shown below. This matrix will be referred to as coefficient matrix ($\boldsymbol{C}_e$).

$$\begin{bmatrix} 0 & 0 & -\frac{1}{C} & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{L} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -R_1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -R_2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ i_1 \\ i_2 \\ \dot{v}_c \\ \dot{i}_L \\ v_c \\ i_L \\ S_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The auxiliary variables in X can be eliminated by taking the reduced row echelon form of $\boldsymbol{C}_e$. Assuming $R_1 = 1$, $L = 5$, $C = 4$, and $R_2 = 1$, the reduced form of above equation can be rewritten as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0.3333 & -0.6667 & -0.3333 \\ 0 & 1 & 0 & 0 & 0 & -0.3333 & 0.6667 & -0.6667 \\ 0 & 0 & 1 & 0 & 0 & 0.3333 & 0.3333 & -0.3333 \\ 0 & 0 & 0 & 1 & 0 & 0.0833 & 0.0833 & -0.0833 \\ 0 & 0 & 0 & 0 & 1 & -0.0667 & 0-1333 & -0.1333 \end{bmatrix} \begin{bmatrix} v_1 \\ i_1 \\ i_2 \\ \dot{v}_c \\ \dot{i}_L \\ v_c \\ i_L \\ S_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This separates the auxiliary variables and gives them as linear differential equations of state variables and inputs. In order to achieve this, there should be $n_s + n_a$ number of equations, where n_s is the number of states, and n_a is the number of auxiliary variables.

The last n_s equations of the linear system give the state equations given below:

$$\dot{v}_c + 0.0833v_c + 0.0833i_L - 0.0833S_e = 0$$
$$i_L - 0.0667v_c + 0.1333i_L - 0.1333S_e = 0.$$

Hence, the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices of the state-space model will be

$$\mathbf{A} = \begin{bmatrix} -0.0833 & -0.0833 \\ 0.0667 & -0.1333 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.0833 \\ 0.1333 \end{bmatrix}$$

for a state vector of $[v_c\ i_L]^T$.

The output equations can also be obtained using the reduced $\boldsymbol{C}_e$ matrix. If the output is v_1, the output equation corresponds to the first row of the set of equations as shown below:

$$v_1 + 0.3333v_c - 0.6667i_L - 0.3333S_e = 0.$$

Then, the $\boldsymbol{C}$ and $\boldsymbol{D}$ matrices will be

$$\mathbf{C} = [-0.3333 \quad 0.6667]$$

$$\mathbf{D} = [0.3333]$$

If the output variable is not in the vector $\boldsymbol{X}$, it can be indirectly obtained by multiplying with some other component values. If the voltage across R_2 is required, multiplying the equation corresponds to i_2 by R_2 will give that quantity.

10.2.2 Linear Graphs

Linear graphs (LGs) use interconnected line segments called branches to represent ideal lumped parameter elements. Each branch has an ordered pair of variables associated with it: a *through variable* (f) and an *across variable* (v) [1–3,9,10]. The product of the two variables is the power variable. Every branch is oriented, and the branches are connected at nodes. One end of a branch is the *point of action*, and the other end is the *point of reference*. An arrowhead pointing toward the *point of reference* is used to describe the positive direction of flow as shown in Figure 10.6.

The value of the *through variable* (f) does not change through an element. The value of the *across variable* (v) of the element is measured at the point of action relative to the point of reference. Table 10.4 presents the typical through and across variables associated with some common domains.

In a linear graph, there are four types of elements: single-port passive elements, single-port active elements, and two two-port elements. Single-port passive elements are C (across-type or A-type energy storage elements), I (through-type or T-type energy storage elements), and R (dissipation-type or D-type elements), which, in the electrical domain, represent capacitors, inductors, and resistors and, in the mechanical domain, represent

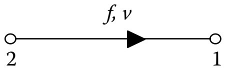


FIGURE 10.6
LG branch of an element.

TABLE 10.4
Through and Across Variables Associated with Several Domains

Domain	Through Variable	Across Variable
Electrical	Current	Voltage
Mechanical	Force/torque	Velocity/angular velocity
Hydraulic/pneumatic	Flow rate	Pressure
Thermal	Heat transfer	Temperature

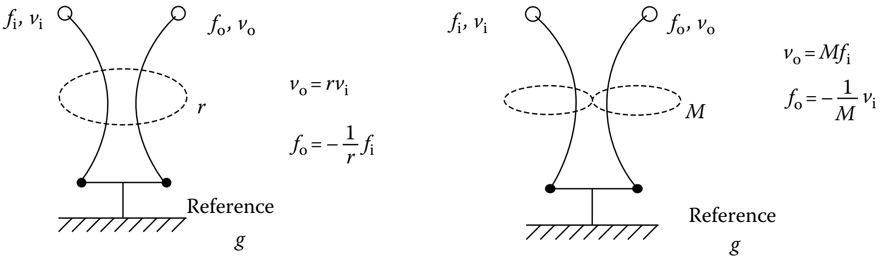


FIGURE 10.7
LG representation of a transformer and a gyrator.

inertia (mass), spring, and damper, respectively [2]. Single-port active elements are source elements, which may be classified as the *T-type source* (current source, force source) and *A-type source* (voltage source, velocity source).

There are two two-port elements, representing transformers and gyrators, in which power is conserved. The *transformer* element is an ideal element in which the across and through variables associated with one branch are transformed into another pair of values without dissipating or storing energy. Figure 10.7 shows the linear graph representation of a *transformer* in which v_i and f_i are the across and through variables of the input port, and v_o and f_o are the across and through variables of the output port. For the transformer, r is the transformation ratio between the two across variables of the branches (and equivalently, for the corresponding two through variables). For the gyroscope, M is the parameter that relates the through variable of one branch to the across variable of the other branch.

Figure 10.8 shows the LG representations of a simple mechanical system in parallel configuration and an electrical system in series configuration.

In order to obtain an analytical model from an LG, three types of equations have to be established [3,10].

- Constitutive (physical) equations
- Compatibility (or loop) equations
- Continuity (or node) equations

The constitutive equations correspond to the "physical" equations of all the branches (elements) except for sources (inputs). The compatibility equations correspond to the independent loops in the LG. Figure 10.8 shows three independent loops formed by branches *f-m*, *m-b*, and *b-k*. There are other loops that can be formed by branches, such as *f-b*, *m-k*, and *f-k*, but those are not independent as they can be derived from the earlier chosen set. However, if one selected the latter set as the independent set of loops, the former set can be obtained by algebraic manipulations. The compatibility equations are obtained by summing all the across variables along a branch of an independent loop to zero. Compatibility assures system integrity in that, at a given point in the system, the across variable value is unique at a given instant. A sign convention dictates that the direction of the branch arrow is the direction of drop of the across variable, which is considered positive except for a T-source in which it is the opposite. Continuity equations are obtained by summing all the through variables at each node of the LG to zero. These equations represent such conditions as force balance at a point in a mechanical system, current balance at a point in

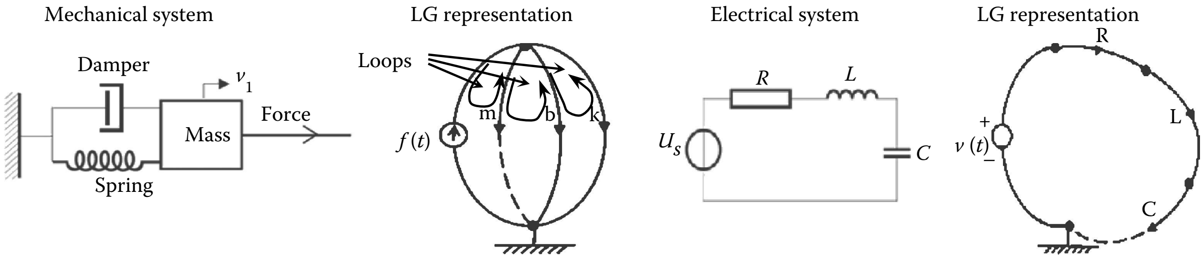

Mechanical system
Damper
v_1
Mass
Force
Spring
LG representation
Loops
m
b
k
$f(t)$
Electrical system
R
L
U_s
C
LG representation
R
L
+
$v(t)$
−
C

FIGURE 10.8
LG representation of a mechanical oscillator and an electrical circuit.

an electrical system, and flow continuity at a point in a hydraulic system. For example, for the LG of the electrical circuit, the following equations can be written.

Constitutive	Node	Loop
$V_R = Ri_R$	Node 1: $i - i_R = 0$	$V(t) - V_R - V_L - V_C = 0$
$V_L = L\dfrac{di_L}{dt}$	Node 2: $i_R - i_L = 0$	
$i_L = C\dfrac{dV_c}{dt}$	Node 3: $i_L - i_C = 0$	

Once the constitutive, compatibility, and continuity equations are written, the systematic methodology to obtain the state-space model comprises the following steps:

1. Select the state variables for the independent elements: across variables for the independent A-type elements and through variables for the independent T-type elements.
2. Write the state-space shell. The state-space shell is formed by the constitutive equations of the independent energy storage elements.
3. Write the remaining constitutive equations for the other elements.
4. Write the continuity equations for the primary (i.e., independent) nodes.
5. Write the compatibility equations for the primary (i.e., independent) loops.
6. Finally, eliminate all the other variables in the state-space shell that are not state variables or input variables, using the node, loop, and remaining constitutive equations.

An illustrative example to demonstrate this methodology is given later in this chapter.

10.3 Design Evolution

It is possible to evolve a design represented by a BG or a LG using evolutionary programming techniques, such as genetic programming. LGs as opposed to BGs provide a realistic representation of the physical structure of a mechatronic system [1,3]. BGs use causality strokes to indicate the causality of two variables associated with each bond, and with this, the system order can be determined.

Genetic programming (GP) is a branch of evolutionary computing and is an extension of genetic algorithms (GA). In GA, a solution is represented by a chromosome, typically a sequence of ones and zeros of fixed length, whereas, in GP, a solution is represented by a tree-like structure of variable length and topology allowing a solution to grow. The growth of the solution is achieved by a set of construction functions and terminals, which are added to modifiable sites so as to optimize a fitness function. A population is made up of a number of trees (chromosomes), and evolution is carried out by the genetic operations of selection, crossover, and mutation. A tree is grown from an *embryo model*, which reflects the fundamental specifications of the solution and is common to all solutions.

Evolution of design models of engineering systems using GP allows a design engineer to explore the design space to find an optimal solution in an automated manner. The evolution produces a large number of design candidates, which need to be evaluated to find the most suitable solution. There are issues that need to be addressed before this approach can be used for design improvement and implementation of practical and complex engineering systems. For example, arbitrary evolution of a design model may result in complex designs that are not feasible in practical implementation.

Identification of independent regions, subsystems, or components of the system that cause performance degradation or faults is important as it reduces the size of the system to deal with and hence of the complexity of the evolution process. This may be achieved using domain knowledge and expertise. The design engineer can focus on the identified regions, subsystems, or components of the system that need to be modified to achieve the desired performance.

10.3.1 Evolutionary Design Framework with BGs

The framework of evolutionary design originates with the combination of BG modeling of dynamic systems with the evolutionary capability of GP to achieve a model-based approach for automated evolution of an engineering design [11–16]. As mentioned earlier, BGs are graphical representations of lumped parameter models of dynamic systems, which provide a unified approach to modeling, and analysis of multidomain systems. BG models describe the dynamic behavior of a system based on the principles of energy and power.

In GP, a solution is represented by a tree-like structure of variable length and topology allowing a solution to grow [17]. The growth of the tree is achieved by a set of construction functions and terminals, which are added to modifiable junctions or bonds as shown in Figure 10.9. For example, the *Add-R* function, which can be applied to a modifiable junction, adds a modifiable *R* element, a modifiable bond, and retains the original modifiable junction. The *Insert-J0* function, which can be applied to a modifiable bond, creates a modifiable "0" junction and two modifiable bonds. Table 10.5 shows the common construction functions and terminals used. A population is made up of a number of trees (chromosomes), and evolution is carried out by selection, crossover, and mutation. A tree is grown from an embryo solution, which reflects the fundamental specifications of the solution and is common to all solutions. Figure 10.10 illustrates how a BG can be grown from an embryo solution by adding construction functions and terminals to modifiable junctions and bonds. As an example, consider the embryo model shown in Figure 10.10a. This embryo model has two modifiable junctions and a modifiable bond. If the GP tree shown in Figure 10.10c is applied to the embryo model, first the construction function *ADD-I* is applied to the modifiable "1"

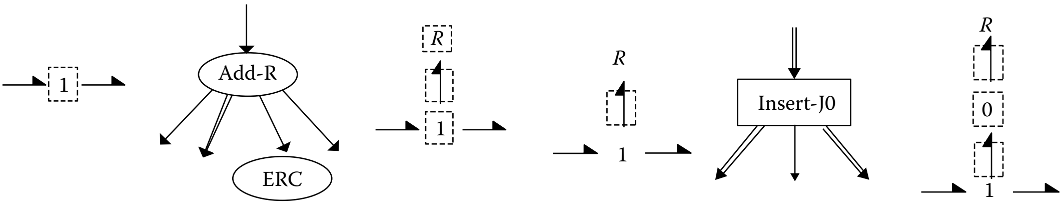


FIGURE 10.9
Growth of a BG from a junction and from a bond.

TABLE 10.5

Construction Functions and Terminals

Name of Construction Function/Terminal	Description
ADD R	Adds an R element
ADD I	Adds an I element
ADD C	Adds a C element
Insert J0	Inserts a 0 junction
Insert J1	Inserts a 1 junction
ERC	Ephemeral random constant
End	Ends further growth

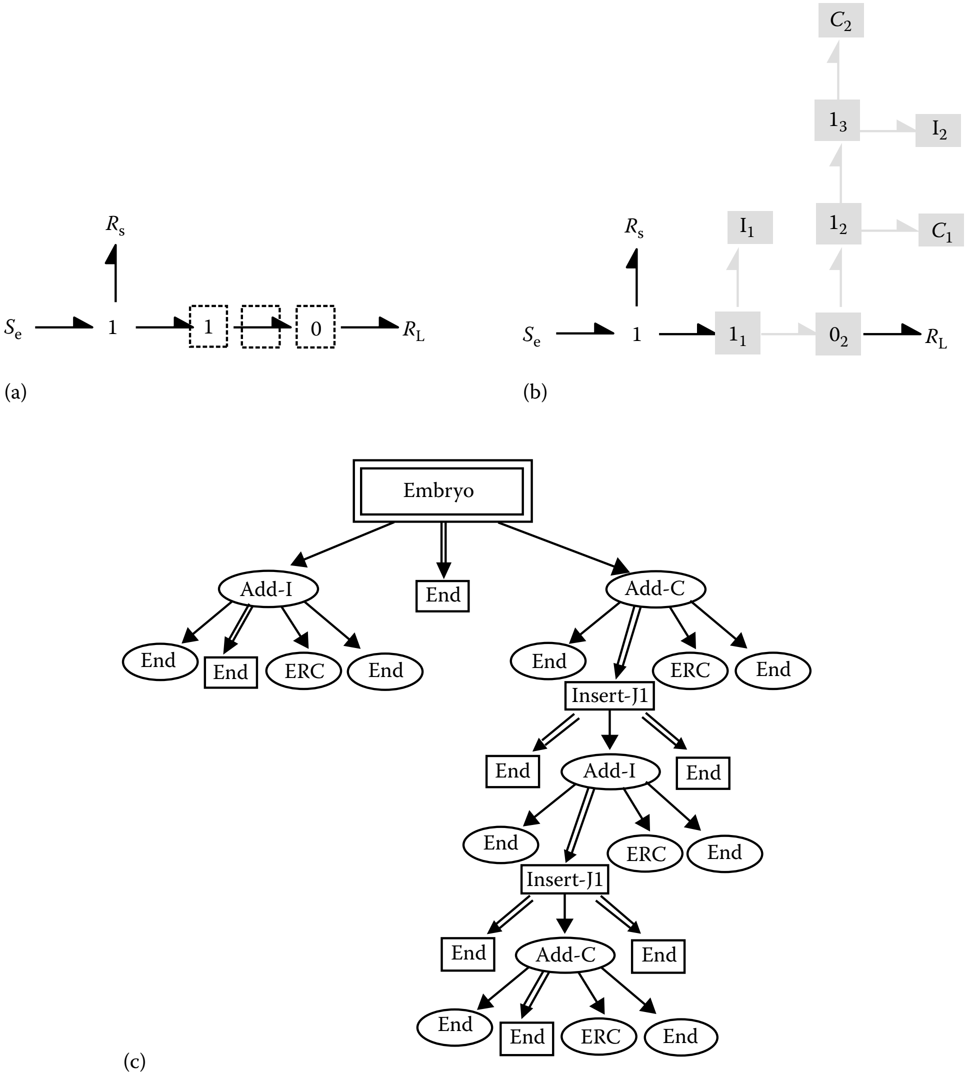

R_s
S_e
1
1
0
R_L
(a)
C_2
1_3
I_2
I_1
1_2
C_1
R_s
S_e
1
1_1
0_2
R_L
(b)
Embryo
Add-I
End
Add-C
End
End
ERC
End
End
Insert-J1
ERC
End
End
Add-I
End
End
Insert-J1
ERC
End
End
Add-C
End
End
End
ERC
End
(c)

FIGURE 10.10

Illustration of BG growth from an embryo model. (a) Embryo BG model with modifiable sites. (b) Evolved BG model. (c) GP tree for the evolution.

junction. This adds a modifiable *I* element with a modifiable bond, retaining the original modifiable junction as modifiable. The attributes specified with the construction functions determine how the growth proceeds. In this example, the attributes of the *ADD-I* function specify that further growth from the original modifiable site and the growth from the created modifiable bond should be terminated (ended). They also specify that the modifiability of the added *I* should be ended after computing its value using ERC.

The next branch attached to the embryo specifies that the growth from the modifiable bond should be terminated. The third branch attached to the embryo first adds a *C* element, and further growth is allowed through the added modifiable branch. This process continues until further growth is not allowed by the terminal functions, and the BG shown in Figure 10.10b is evolved. The evolution of the population in this open and structured manner allows us to explore the design space and generate design candidates in an effective way.

10.3.2 Methodology

As mentioned before, a multidomain engineering system design procedure can be considered as finding the optimum topology of the system and the parameter values of its components. As an example, in an analog filter design problem, the same set of resistors, inductors, and capacitors can be organized to a circuit in numerous ways. Of these possibilities, the design engineer must choose the set of connections (topology) that gives the required performance and also the corresponding numerical values of the components, such as capacitors, resistors, and inductors. The main parts of the design process are representation of the engineering system using BG or LG modeling, evolution of the model using GP, and evaluation of the evolved model to check its fitness with respect to user-defined requirements.

10.3.3 Solution Representation for the Evolution

Because GP employs tree structures for the evolutionary process, the multidomain engineering system has to be represented as a tree structure. In a BG-GP–based approach, the BG model of an engineering system is constructed as a tree. The evolutionary process is started by first having an embryo model of the system. The embryo contains the information about the basic topology, inputs and outputs to the system, and the locations of the system that are modified during the evolutionary process. An embryo of an electrical circuit and its BG representation are illustrated in Figure 10.11. The basic elements used for modeling the engineering systems are capacitors, inductors, resistors, voltage sources, current sources, transformers, and gyrators and the mechanical or hydraulic counterparts of these elements. During the evolution, the embryo is evolved until an optimal (desired) solution is obtained. The modifications include changing the topology of the system by

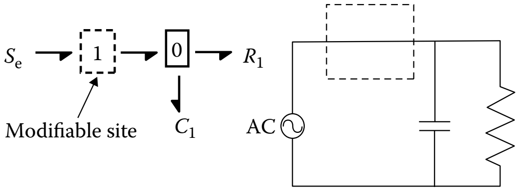


FIGURE 10.11
Embryo model.

adding and/or changing of components and adjusting of parameter values of components at specified modifiable sites. These modifications are carried out by a set of construction functions and terminals given in Table 10.5.

Like all evolutionary schemes, GP also considers a set of candidate solutions, which is called a population during the optimization. Given the embryo model, the initial population is generated by randomly choosing construction functions. A random number is chosen from a uniform distribution, and a construction function is chosen, depending on the range of that random variable. Because a construction function itself can generate more modifiable locations, it is important to have an end of operations construction function in order to stop growing the GP tree indefinitely. Once the complete population is obtained by randomly growing these GP trees, each tree is converted to a BG. The state-space model corresponding to each BG is then established and simulated for the purpose of evaluating the performance (suitability) of the solution. The performance of a given solution is measured by a fitness function, which represents the suitability of the solution.

10.3.4 Fitness Function

The fitness function that is used for design evolution is termed the mechatronic design quotient (MDQ). This concept was proposed by de Silva in 2003 [18,19] and subsequently further enhanced by Behbahani and de Silva [1,3,13,16]. The fitness function for a given design problem depends on the system and the problem itself, but it can be considered as a measure of the closeness of the obtained model solution to the desired solution. For example, for a position control system, the desired response and the model response can be as shown in Figure 10.12. A fitness function for this can be defined as a matching index as given below. The matching index is essentially a measure of the error between the model output and the required output. It is computed as the square root of the sum of error squares (SRSS) and normalized with respect to the number of samples (n) as shown below.

$$\text{Matching index (MI)} = \frac{n}{n + \text{SRSS}}$$

$$\text{SRSS (error)} = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2}$$

The fitness of each individual in a population is calculated, and they are ranked according to the fitness value. After the initial population, subsequent populations are obtained

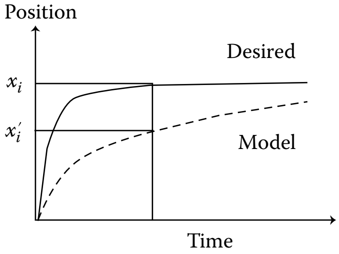


FIGURE 10.12
Desired and model responses of a position control system.

by genetic operations, such as repetition, crossover, and mutation, until the desired (optimum) solution is achieved. Repetition copies the chromosomes in the parent population to the child population without any alteration. In the crossover operation, two branches of the GP tree of two different parents are chosen, and they are interchanged to produce two new offspring. Mutation creates a new individual by adding randomly created branches to the GP tree. This is useful in achieving the global optimum because this operation will introduce new solutions that have not been considered previously. Because the repetition directly copies parents in their entirety to the child population, it may lead to premature convergence of the GP. Crossover and mutation add diversity to the evolution.

The selection of individuals to these operations is based on their calculated fitness values. Parents having higher fitness values are given a higher probability of selection for genetic operations so that better individuals are obtained in subsequent generations. At each generation, fitness of the solutions is calculated to decide whether an individual with desired fitness has been evolved. This loop is continued until the optimal system is obtained as illustrated in Figure 10.13.

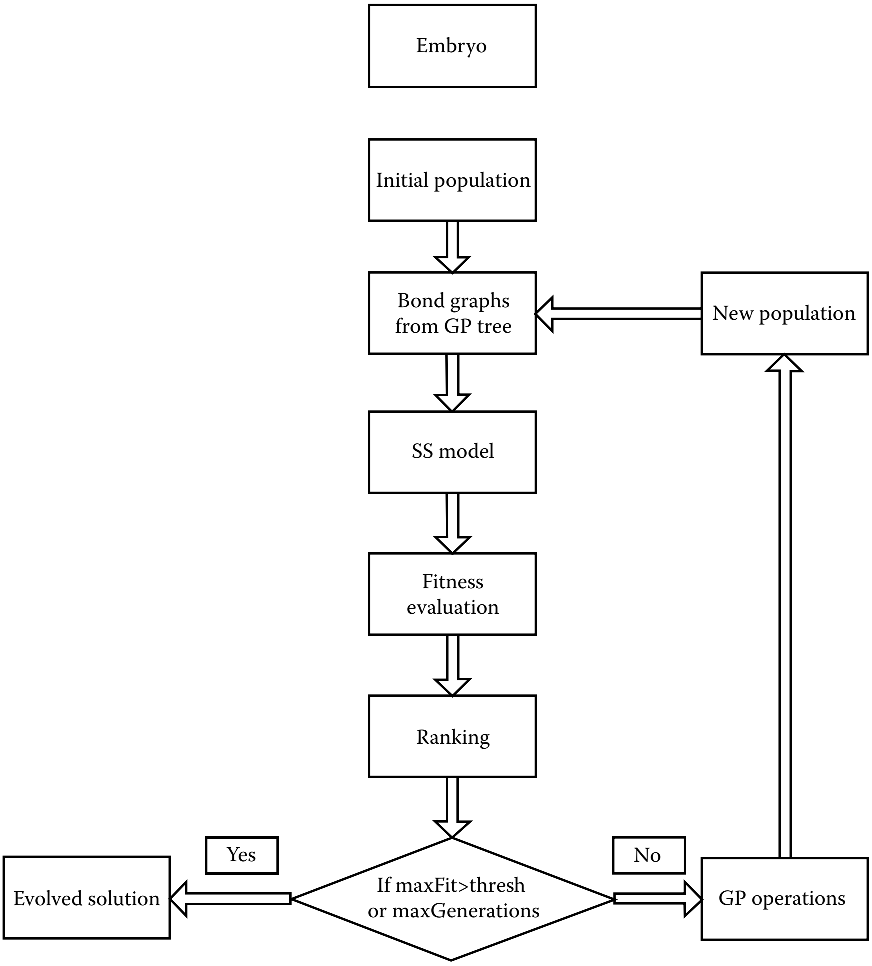

Embryo
Initial population
Bond graphs from GP tree
New population
SS model
Fitness evaluation
Ranking
Yes
No
Evolved solution
If maxFit>thresh or maxGenerations
GP operations

FIGURE 10.13
BG-GP flowchart.

Example 10.2: Evolution of a Mechanical System

In this example, we look at how a system can be evolved to satisfy a given transfer function. The embryo model used for this problem is given in Figure 10.14a. Input to the system is the force applied to mass, m_1, and output is the velocity of mass, m_2. The desired transfer function is

$$\frac{V(s)}{F(s)} = \frac{4s + 40}{s^3 + 40s^2 + 400s}$$

For the fitness calculation, absolute error between the frequency responses of desired transfer function and the candidate system's transfer function is used. Samples of the response at 15,000 locations in a frequency range of $1\text{–}10^4$ rad/s in a logarithmic scale were obtained, and the fitness values were calculated as follows:

$$f = \frac{1}{1 + E[|\text{error}|]}$$

Where $E[|\text{error}|]$ is the average absolute error between the desired and evolved system frequency responses. A population size of 60 individuals was used for this simulation, and the final solution was obtained after 20 generations. Figure 10.14b and c show the evolved system and the fitness improvement during the evolution.

Example 10.3: Evolution of an Electrical System

In this example, a band-pass filter with a pass band of $5 \times 10^4 - 10^5$ rad/s was evolved to satisfy the transfer function, which was obtained with the Butterworth approximation given below. Fitness was computed as given in the previous example.

$$H(s) = \frac{5 \times 10^4 s}{s^2 + 5 \times 10^4 s + 5 \times 10^9}$$

The evolved band-pass filter using the described method is shown in Figure 10.15.

10.4 Application of Methodology to Industrial Systems

A fully automated machine called the "intelligent iron butcher" or IIB shown in Figure 10.16 has been designed and developed for the fish processing industry by the Industrial Automation Laboratory of the University of British Columbia [20]. The purpose of the machine is to accurately remove the heads of Pacific salmon at high speed. The conventional machines used in the industry waste about 10%–15% of useful meat with each unit percentage of wastage costing about $5 million annually in the province of British Columbia, Canada. The IIB consists of two main subsystems, the cutter assembly with its positioning system and the conveying system. Both these subsystems can be considered as mechatronic systems synchronized to work together as one system. The positioning system employs hydraulic actuators, servo valves, and a computer vision system to realize

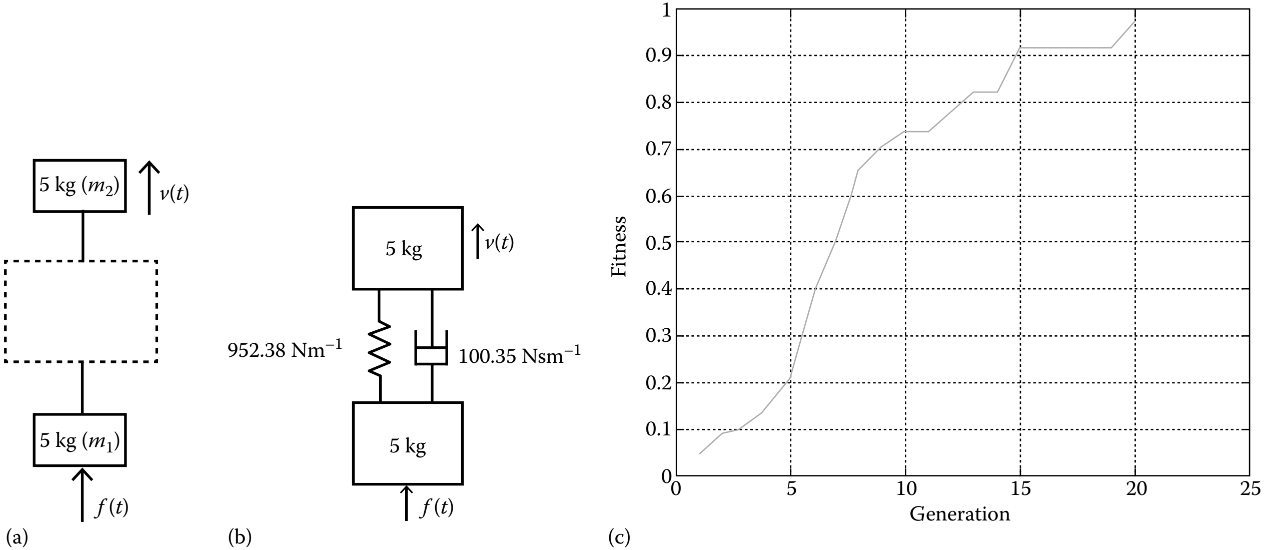


FIGURE 10.14
Evolution of a mechanical system. (a) Embryo model. (b) Evolved system. (c) Fitness evolution.

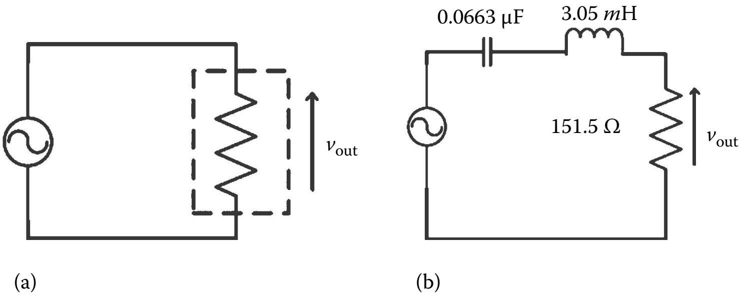


FIGURE 10.15
Band-pass filter evolution. (a) Embryo model and (b) final solution.

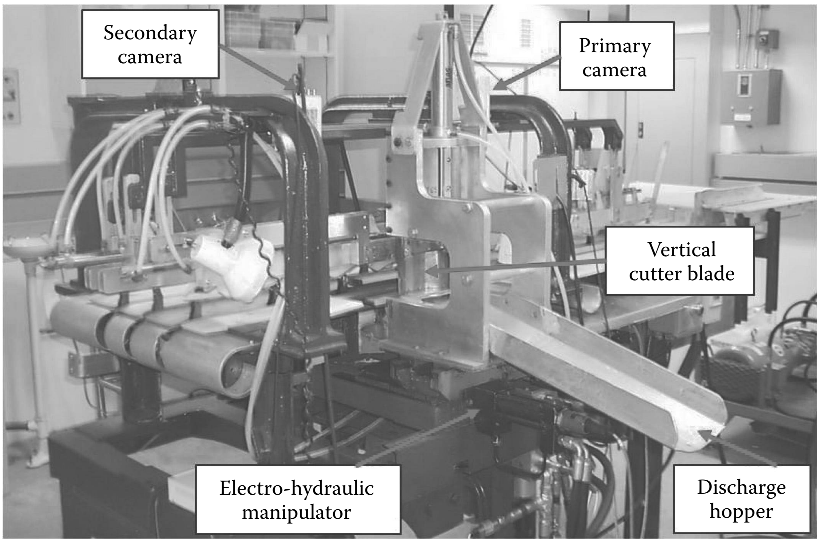


FIGURE 10.16
Intelligent iron butcher.

accuracies of the order of 1%. In the IIB operation, the vision system captures an image of the head section of the fish on the conveyor to be processed, and its gill position is computed. A cutter assembly fixed on an x-y positioning table is aligned with the gill position according to the computed value so that the fish head can be removed with minimal wastage. This entire operation, including the computation of the gill position by the vision system, has to be carried out within 500 ms, satisfying the required feed rate of two fish per second. Another computer vision system captures images of fish immediately after the head removal operation to determine the quality and the accuracy of the cut. The conveying system, driven by an AC induction motor, moves the fish forward in an intermittent manner so that the fish is kept stationery when the image is captured and also when the head is removed. A pneumatically operated cutter, which is formed in the shape of an average gill, is pushed down to remove the head along the collarbone. The positioning table, including the cutter assembly, is actuated by two electrohydraulic manipulators controlled by two servo valves.

In order to monitor the performance and the condition of the IIB, a health monitoring system (HMS) is attached to it as shown in Figure 10.1. The HMS obtains signals from a large number of sensors mounted on the machine (engineering system). They include accelerometers, position sensors, pressure sensors, strain gauges, microphones, and cameras. The signals obtained from these sensors are processed to monitor the system performance and to determine whether there are any faults. An expert system (ES) as shown in Figure 10.1, extracts information from the HMS and makes a decision as to whether a design improvement is necessary. The information obtained from the HMS includes the level of performance degradation (compared to specified requirements), frequency of a particular fault, the location and the nature of the fault, and diagnostics.

If information from the HMS indicates a performance issue and if the ES determines that this issue is due to a design weakness, a design improvement has to be made. The reason for this may be due to the system not performing as specified with respect to a set of specifications or a regular occurrence of a malfunction or a fault possibly due to a design weakness. The ES isolates the region of design weakness with the assistance of the HMS.

The methodology of design evolution described above is applied for the purpose of design improvement of the machine. First, a model of the overall system is developed in which the region of design weakness is clearly identified. The parameter values for the model are obtained from the information on the original prototype system, available original design documents, and if necessary through experimentation (i.e., experimental modeling, system identification). Once the model is developed, a simulation is carried out to check whether the system performs according to the specified functional requirements. Functional requirements of the system, for example, can be specified as a set of expected responses in the time domain and/or the frequency domain. If the simulation results do not meet the specified functional requirements, after ruling out component failures, it is determined that there exists a design problem.

Identification of the system model (both parameter values and the structure) of the implemented system is carried out using the methodology described in Section 10.4.1.2. This methodology uses a BG-GP approach in which a BG embryo model of the system is evolved by GP until it satisfies a fitness function.

In the evolutionary design approach, it is necessary to identify the locations of the nodes (junctions), bonds, and the parameters (values) of the BG model that will be modified by evolution using the GP. These modifiable sites need to be carefully chosen with the help of the ES (knowledge, expertise, and experience) to restrict the GP exploration only to a plausible design space. Once the embryo model is developed in this manner, it is evolved until the desired functional requirements are obtained. The functional requirements therefore form the fitness function for the design evaluation.

There can be several solutions that satisfy the specified functional requirements. However, the possibility of producing infeasible solutions, to a great extent, is avoided by incorporating available knowledge (through the ES) and careful selection of modifiable sites. The selection of the final design solution will depend on such factors as its complexity and cost of implementation, which are included in the fitness function. Once a design solution is established, it will be implemented, and its performance will be observed to make sure that the functional requirements are met. If they are not satisfied, corrective actions (possibly through tuning and control) may have to be taken, or, in the worst case, the steps of design evolution may have to be repeated until the desired results are obtained. Clearly, repetition of design implementation can be costly. Therefore, the ES needs to be as intelligent as possible to determine the feasible designs. The flowchart of the algorithm of design evolution, which, in this case, employs BGs for modeling, is shown in Figure 10.17.

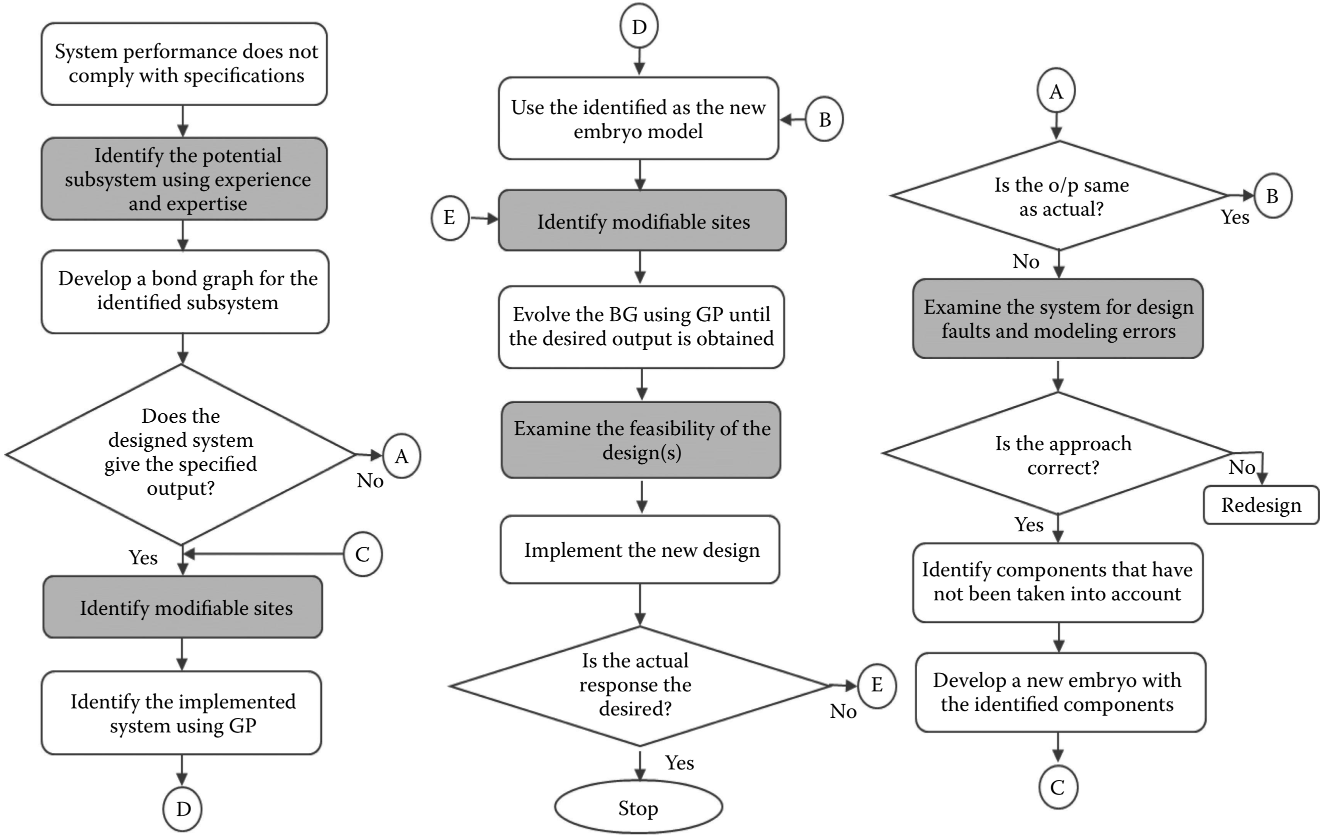

System performance does not comply with specifications
Identify the potential subsystem using experience and expertise
Develop a bond graph for the identified subsystem
Does the designed system give the specified output?
No
A
Yes
C
Identify modifiable sites
Identify the implemented system using GP
D
D
Use the identified as the new embryo model
B
E
Identify modifiable sites
Evolve the BG using GP until the desired output is obtained
Examine the feasibility of the design(s)
Implement the new design
Is the actual response the desired?
No
E
Yes
Stop
A
Is the o/p same as actual?
Yes
B
No
Examine the system for design faults and modeling errors
Is the approach correct?
No
Redesign
Yes
Identify components that have not been taken into account
Develop a new embryo with the identified components
C

FIGURE 10.17
Design evolution algorithm.

The highlighted blocks of this flowchart receive information from the HMS or the ES to carry out their functions.

10.4.1 Illustrative Scenario 1

In the present example, the HMS of the fish-cutting machine observes that fish is not cut at the correct gill position from time to time. Specifically, the cutter assembly does not move to the correct position within the specified time as shown in the response curve of Figure 10.18. This may be primarily due to the following two reasons: The vision system does not compute the gill position correctly or the hydraulic cutter positioning system does not perform properly. A block diagram of the hydraulic system of the fish cutting machine is shown in Figure 10.19, and schematic diagrams of a servo valve and a hydraulic actuator are shown in Figure 10.20a and b, respectively.

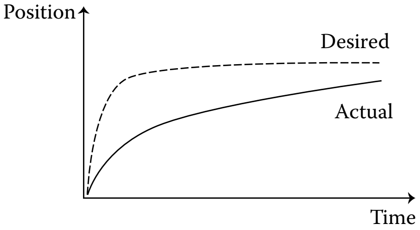


FIGURE 10.18
Cutter assembly response.

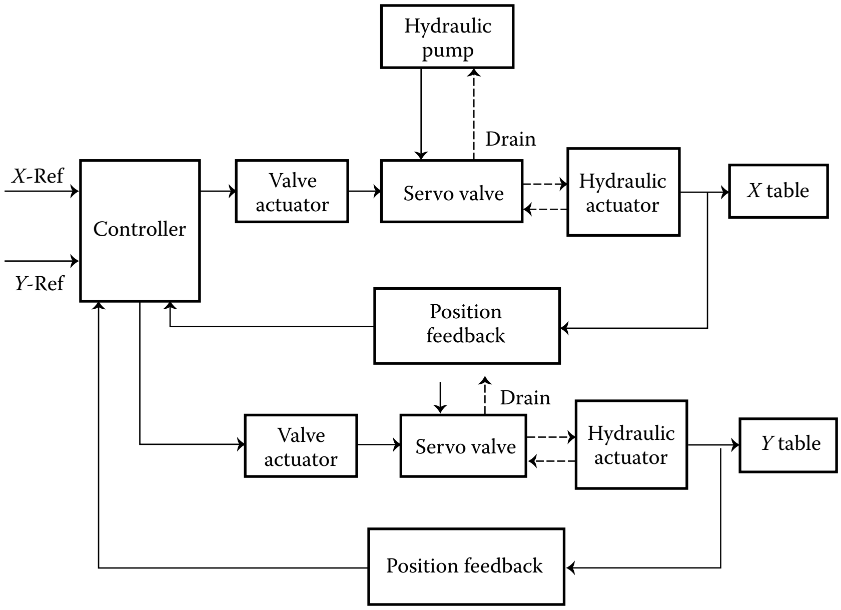

Hydraulic pump
Drain
X-Ref
Controller
Valve actuator
Servo valve
Hydraulic actuator
X table
Y-Ref
Position feedback
Drain
Valve actuator
Servo valve
Hydraulic actuator
Y table
Position feedback

FIGURE 10.19
Block diagram of the hydraulic system.

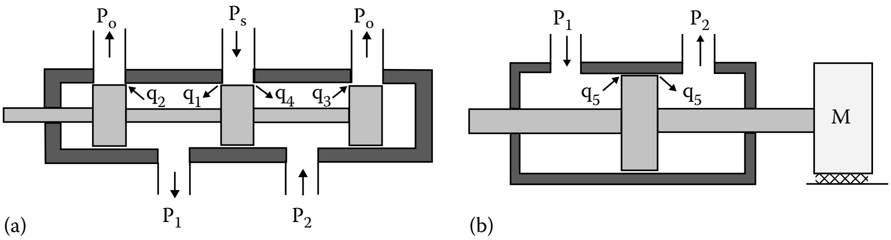


FIGURE 10.20
(a) Servo valve and (b) hydraulic actuator.

Further investigation by the ES, particularly using information on past performance, rules out a fault in the vision system. Further analysis by the HMS using data obtained from the machine pressure sensors and a trained Kalman filter reveals that the performance degradation is due to a pressure drop (leakage) in the servo valve (X direction). At this stage, a decision has to be made whether to replace the servo valve, if this type of leakage is common in the servo valves after a certain period of operation, or make improvements to the machine design in the region of the cutter positioning system. It should be noted that at the time of the design this leakage was not anticipated and not considered in the original design process. Therefore, a model developed for the original design would result in acceptable specified performance. Referring to Step 4 of the algorithm given in Figure 10.17, this is a case in which the designed model gives the correct output in simulation, and the actual system does not. Hence, the implemented prototype system must be correctly identified (modeled) with the intension of making improvements to the design.

10.4.1.1 Fitness Function

The fitness function used for both system identification and design evolution has three main components. The first component, the matching index (MI), examines the degree of closeness to the required output from the system. In the case of system identification, the required output is the actual output obtained from the implemented prototype system as shown in Figure 10.21a. For design evolution, the required output is the specified, desired output of the system as shown in Figure 10.21b.

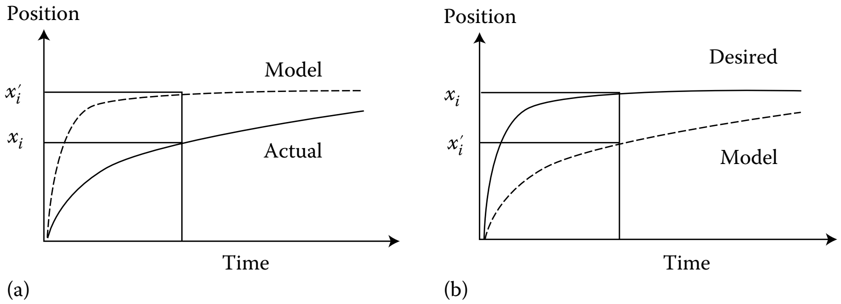


FIGURE 10.21
Responses for (a) system identification and (b) design evolution.

The MI is essentially the error between the model output and the required output. It is computed as the SRSS and normalized with respect to the number of samples (n) as given by

$$\text{SRSS (error)} = \sqrt{\sum_{i=1}^{n} \left(x_i - x_1'\right)^2} \quad \text{Matching index (MI)} = \frac{n}{n + \text{SRSS}}$$

The design evolution using GP may result in more than one solution that can meet the specified performance requirements. However, some solutions may be more complex than others in terms of the number of components and their interconnections (topology). Therefore, it is desirable to select the least complex design solution that meets the specified requirements. The second component of the fitness function is the simplicity index (SI), which deals with this issue and is defined as

$$\text{Simplicity index (SI)} = \frac{\text{(Number of modifiable sites)}}{\text{(Number of added nodes and bonds)}}$$

This ratio will be one if only the introduced modifiable sites are changed, and it will become smaller as the complexity increases. In most practical situations, it is desirable to achieve the required performance through as few changes as possible. For example, if a particular solution requires the hydraulic pressure to be increased substantially, requiring new components such as a pump and its accessories, the cost and the required effort of this solution would be higher than a solution that requires the pressure to be increased to a value that is manageable by the existing components. The second solution may, however, require some other components to be changed as well, but they may also be within their current capacities or may not be as costly and time-consuming as the installation of a new pump. To select such preferred solutions over those with large parameter variations, a third index called the parameter variation index (PI) is incorporated into the fitness function, as defined by

$$\text{Parameter variation index (PI)} = \frac{\sum_{i=1}^{m} \left| \frac{p_i - p_i'}{p_i} \right|}{m}$$

Here, p_i is the current parameter value, p_i' is the changed parameter value, and m is the number of changed parameters. Combining all three indices, the fitness function is defined as

$$\text{Fitness} = \alpha\text{MI} + \beta\text{SI} + \gamma\text{PI}$$

Here α, β, and γ are weighting factors, which are chosen depending on the problem. In particular, in the case of system identification, a low value for γ must be selected to allow the model to evolve to a true representation of the implemented prototype system.

10.4.1.2 System Identification

For the purpose of system identification, an embryo BG model was developed based on the original design of the hydraulic cutter positioning system. To avoid unnecessary free evolution of the BG model using GP, the appropriate sites of the embryo model are chosen for modification with the help of the information obtained from the HMS. In the present example, the HMS has indicated that the performance degradation is due to a leakage in the servo valve. In order to model the system with the leakages, four modifiable sites are introduced to the developed embryo model as shown in Figure 10.22.

The GP-based evolution program is executed with these modifiable sites and with the appropriate weighting parameters in the fitness function until the fitness value of a solution reaches the specified threshold value. The fitness value represents not only the closeness of the model output to the actual output but also the complexity (or simplicity) of the solution. The evolution programs are written and executed using MATLAB, and the simulations are carried out using Simulink®. The weighting parameters are chosen on a trial-and-error basis.

The system model was identified with the chosen modifiable sites and weighting parameters. The changes can be seen as R_1', R_2', R_3', and R_4' in the servo valve section of Figure 10.23. This model is an accurate representation of the implemented prototype system, which gives the unsatisfactory output (performance). Therefore, now this model needs to be improved to give the desired output. As before, appropriate modifiable sites are determined by the ES using available knowledge on the problem (poor performance), and the type of design modifications that can be afforded (perhaps before a major overhaul can be carried out) are chosen. In this example, initially, changes to the power source (pressure) and modifications to the cutter are prescribed. In particular, the cutter weight and its resistance to motion are allowed for remodeling. If the system performance cannot be improved to a satisfactory level with these changes, further modifications may be allowed, depending on their feasibility and cost.

The design evolution is carried out with GP with the identified model of the currently implemented model shown in Figure 10.23 as the embryo. Figure 10.24 shows the identified modifiable sites: the input pressure for the servo valve and the weight and the friction of the cutter positioning assembly. This model was evolved, and values for S_e, I_1, and R_8 were determined, and the positioning system was modified with the recommended values.

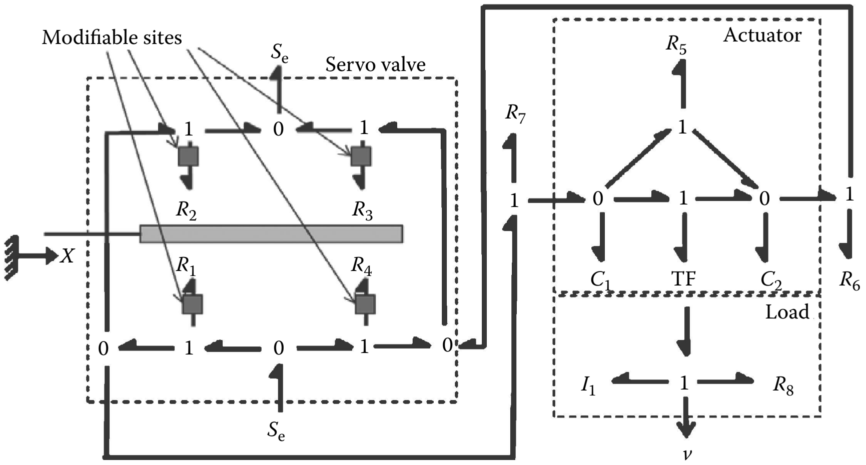


FIGURE 10.22
Embryo model with modifiable sites in the servo valve.

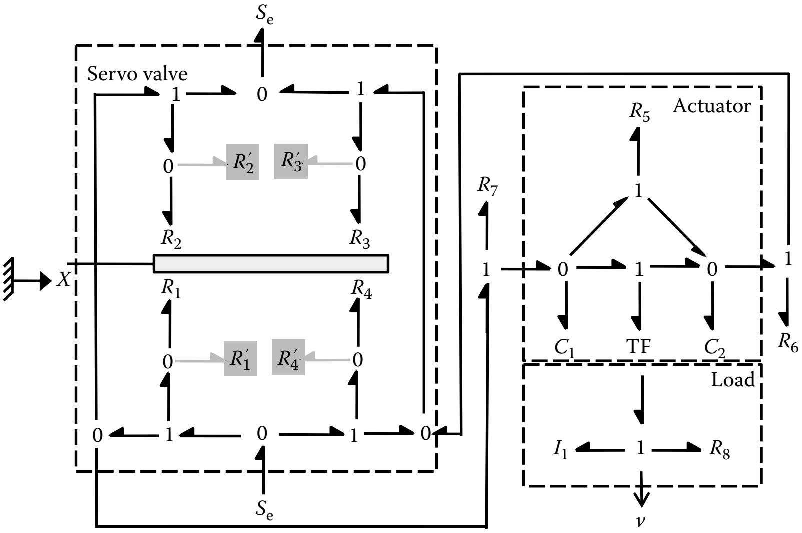

S_e
Servo valve
1
0
1
0
R_2'
R_3'
0
R_2
R_3
X
R_1
R_4
0
R_1'
R_4'
0
0
1
0
1
0
S_e
R_7
1
Actuator
R_5
1
0
1
0
1
C_1
TF
C_2
R_6
Load
I_1
1
R_8
v

FIGURE 10.23
Identified model of currently implemented system.

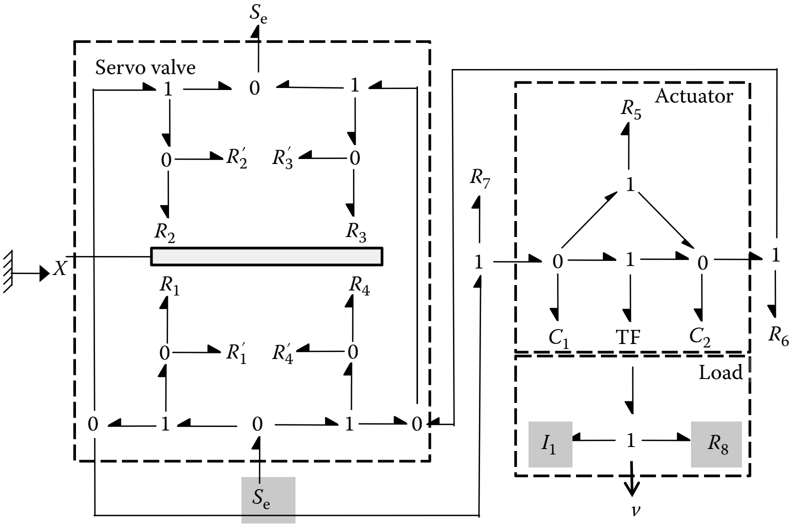

S_e
Servo valve
1
0
1
0
R_2'
R_3'
0
R_2
R_3
X
R_1
R_4
0
R_1'
R_4'
0
0
1
0
1
0
S_e
R_7
1
Actuator
R_5
1
0
1
0
1
C_1
TF
C_2
R_6
Load
I_1
1
R_8
v

FIGURE 10.24
Model for evolution with pressure source and the weight and friction as identified modifiable sites.

10.4.2 Illustrative Example of Application of LG Methodology

In this example, we will illustrate how LG methodology could be applied to improve the design of the electromechanical conveying system of the IIB [7]. The conveying system is driven by an induction motor coupled with a variable diameter pulley (VDP) drive as shown in Figure 10.25. The VDP compensates for speed variations of the motor or load as the distance between the pulleys and the length of the belt does not change. However,

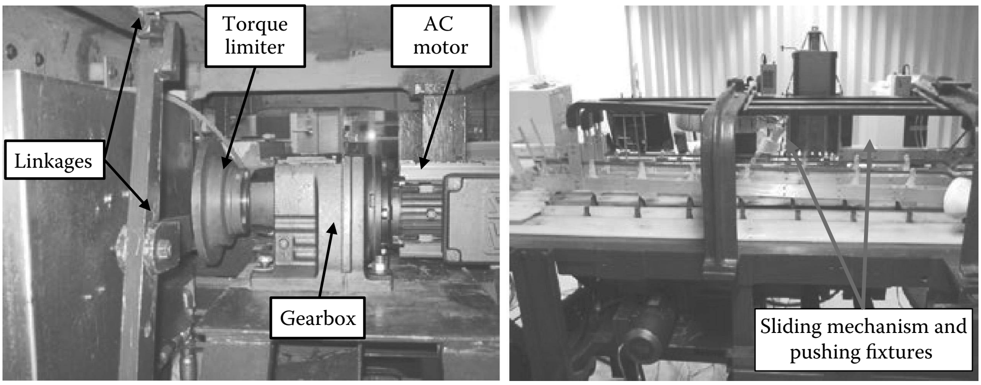


FIGURE 10.25
IIB conveying system with sliding mechanism.

the effective diameter of the pulleys changes, which, in turn, adjusts the overall transmission ratio. The VDP is connected to a gearbox, which decreases the output velocity and increases the output torque. The resulting rotary motion is converted into a push–pull stroke through a mechanical linkage. The mechanical linkage is connected to a sliding mechanism (Figures 10.25 and 10.26), which moves the fish only in one direction as a set of fingers fold or remain open depending on the cycle of the motion.

10.4.2.1 Modeling and Simulation of the Conveying System

A lumped-parameter LG model for the electromechanical conveying system of the IIB is developed as shown in Figure 10.26.

The next step is to derive the state-space model for this dynamic system. According to the methodology described in Section 10.2.2, the following are established for the LG model of the conveying system: A list of parameters defined for the subsystems (gearbox, wheel and axle, and lever) that form the electromechanical conveying system is given below.

1. Gearbox

   ω_1 = input shaft angular velocity, ω_2 = output shaft angular velocity, T_1 = torque on the input shaft, T_2 = torque on the output shaft, r_{GB} = gear ratio

2. Wheel/axle

   v_W = linear velocity of the wheel periphery, ω_A = angular velocity of the axle, T_A = torque on the axle, F_W = force on the wheel periphery, r_{AW} = wheel radius

3. Two-node first-class lever with the fulcrum at node B (Figure 10.30)

   v_1 = lever joint input velocity, v_2 = lever joint output velocity, F_1 = lever joint input force, F_2 = lever joint output force

$$K_{AC} = \frac{l_{BC}}{l_{BC} + l_{AC}} \quad \text{where } l_{AC} \text{ and } l_{BC} \text{ are the arm lengths.}$$

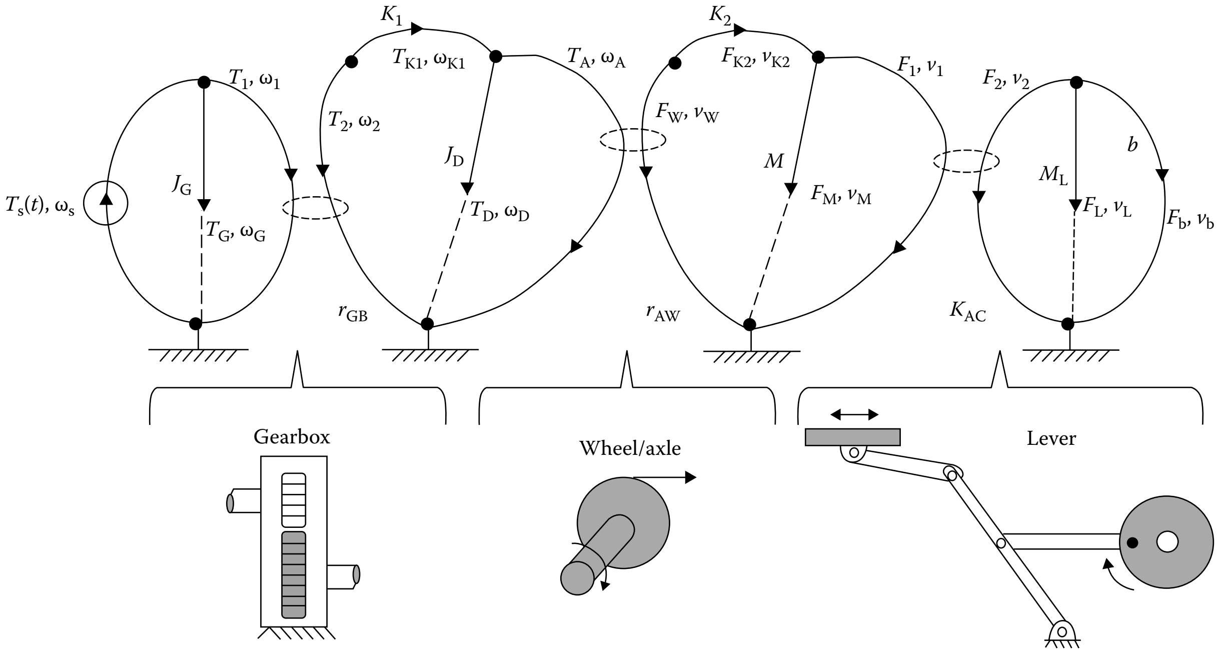

K_1
T_{K1}, ω_{K1}
T_A, ω_A
K_2
F_{K2}, v_{K2}
F_1, v_1
T_1, ω_1
T_2, ω_2
F_W, v_W
F_2, v_2
J_D
M
b
M_L
$T_s(t), \omega_s$
J_G
T_G, ω_G
T_D, ω_D
F_M, v_M
F_L, v_L
F_b, v_b
r_{GB}
r_{AW}
K_{AC}
Gearbox
Wheel/axle
Lever

FIGURE 10.26
LG of the electromechanical conveying system.

The driving motor is represented by an ideal torque source T_S with an angular velocity of ω_S. The following parameters are also defined:

T_G = equivalent moment of inertia of the gear, J_D = equivalent moment of inertia of the wheel (disc), M = equivalent mass of the linkage, M_L = equivalent mass of the sliding copper bars (load), K_1K_2 = stiffness of the "flexible" shaft coupling the gearbox with the disc, K_2 = stiffness of the "flexible" shaft coupling the disc with the linkage, b = equivalent viscous damping constant, v_b = linear velocity of damping element, T_G = torque at the gear, T_{k1} = torque at the "flexible" shaft coupling the gearbox with the disc, T_D = torque at the wheel (disc), ω_G = angular velocity of the gear, ω_D = angular velocity of the wheel (disc), F_{K2} = force at the "flexible" shaft coupling the disc with the linkage, F_M = equivalent dynamic force of the linkage, F_L = equivalent dynamic force of the sliding copper bars (load), F_b = force at the equivalent viscous damper, w_{k1} = angular velocity of the "flexible" shaft coupling the gearbox with the disc, v_m = linear velocity of the linkage, v_{K2} = velocity of the "flexible" shaft coupling the disc with the linkage, v_L = linear velocity of the sliding copper bars.

Further, based on the LG model of the conveying system, the following can be determined:

Number of branches, $b = 14$, number of nodes $n = 7$, number of primary loops $l = 8$, number of sources, $s = 1$.

As each source branch has one unknown variable and all the other passive branches have two unknown variables each, then the number of unknown variables: $2b - s = 28 - 1 = 27$.

Number of constitutive equations: $b - s = 14 - 1 = 13$.
Number of independent node equations: $n - 1 = 7 - 1 = 6$.
Number of loop equations: $b - n + 1 = 14 - 7 + 1 = 8$.

Now as verification, for the system to be solvable, the number of unknowns should be equal to the number of equations:

$$2b - s = b - s + n - 1 + l$$

$$27 = 13 + 6 + 8$$

The system modeled by the LG is a sixth-order system because it has six independent energy storage elements. The selected state variables are

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [\omega_G, \omega_D, T_{K1}, v_m, F_{K2}, V_L]^T$$

Hence, the state-space shell is formed by the following equations:

$$\dot{\omega}_G = \frac{1}{J_G} T_G,\ \dot{\omega}_D = \frac{1}{J_D} T_D,\ \dot{T}_{K1} = K_1 \cdot \omega_{K1},\ \dot{v}_m = \frac{1}{M} F_m,\ \dot{F}_{K2} = K_2 \cdot v_{K2},\ \dot{v}_L = \frac{1}{M_L} F_L$$

The remaining constitutive equations are

$$F_b = b \cdot v_b, \quad v_1 = K_{AC} \cdot v_2, \quad F_2 = K_{AC} \cdot F_1, \quad v_W = r_{AW} \cdot \omega_A$$

$$T_A = r_{AW} \cdot F_W, \quad \omega_1 = r_{GB} \cdot \omega_2, \quad T_2 = r_{GB} \cdot T_1$$

The node equations are

$$T_s(t) - T_G - T_1 = 0, \quad -T_2 - T_{K1} = 0, \quad T_{K1} - T_D - T_A = 0$$

$$-F_w - F_{K2} = 0, \quad F_{K2} - F_M - F_1 = 0, \quad -F_2 - F_L - F_b = 0$$

and the loop equations are

$$\omega_S - \omega_G = 0, \quad \omega_G - \omega_1 = 0, \quad \omega_2 - \omega_D - \omega_{K1} = 0, \quad \omega_D - \omega_A = 0$$

$$v_w - v_m - v_{K2} = 0, \quad v_m - v_1 = 0, \quad v_2 - v_L - v_b = 0$$

Then, by eliminating the auxiliary variables in the state-space shell, the following state equations are obtained:

$$\dot{\omega}_G = \frac{1}{J_G}\left[T_G(t) + \frac{T_{K1}}{r_{GB}}\right], \ \dot{\omega}_D = \frac{1}{J_D}[T_{K1} + r_{AW}\cdot F_{K2}], \ \dot{T}_{K1} = K_1\left[\frac{\omega_G}{r_{GB}} - \omega_D\right]$$

$$\dot{v}_m = \frac{1}{M}\left[F_{K2} + \frac{b\cdot v_m}{K_{AC}^2} - \frac{b\cdot v_L}{K_{AC}}\right], \ \dot{F}_{K2} = K_2[r_{AW}\cdot\omega_D - v_m], \ \dot{v}_L = \frac{1}{M_L}\left[\frac{b\cdot v_m}{K_{AC}} - b\cdot v_L\right]$$

These represent the state-space model. The corresponding system matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \frac{1}{J_G\cdot r_{GB}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_D} & 0 & \frac{r_{AW}}{J_D} & 0 \\ \frac{K_1}{r_{GB}} & -K_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{b}{M\cdot K_{AC}^2} & \frac{1}{M} & -\frac{b}{M\cdot K_{AC}} \\ 0 & K_2\cdot r_{AW} & 0 & -K_2 & 0 & 0 \\ 0 & 0 & 0 & \frac{b}{M_L\cdot K_{AC}} & 0 & \frac{b}{M_L} \end{bmatrix};$$

The input distribution matrix $\mathbf{B} = \begin{bmatrix} \frac{1}{J_G} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$;

The measurement gain matrix **C** = [0 0 0 0 0 1]; and **D** = 0.

Simulations can now be performed as the state-space model, which completely defines the system, has been derived from the LG model. The simulations allow the design engineer to determine the system response to external inputs without the need to build a real system or make changes to an existing system. The developed analytical model of the IIB conveying system is validated by comparing simulation results with the actual results obtained from the experimental setup. MATLAB-based Simscape language is used to code the programs for simulation. Simscape is an extension of Simulink software, which allows modeling of a multidomain physical system by using blocks that represent physical components and their connections. Simscape along with its extensions SimMechanics® and SimElectronics® provide the required blocks for modeling and simulation of the electromechanical conveying system of the IIB. Furthermore, the software allows the development and integration of custom developed blocks.

For the purpose of Simscape modeling, the conveying system is divided into two subsystems, the "driving" subsystem and the "driven" subsystem. The power source and the gearbox form the driving subsystem whereas the driven subsystem consists of a wheel and axle mechanism coupled to a three-linkage mechanism.

The power source is considered as an ideal torque source formed by an AC induction motor connected to a VDP. However, for the experimental setup, a variable frequency drive (VFD) shown in Figure 10.27 replaces the VDP. The VFD is used to control the four-pole, 1 hp, AC motor (Figure 10.27a) in a closed loop to maintain a constant torque and a constant speed. The AC motor is directly coupled to a three-stage helical gearbox with a gear ratio of 106.58:1. The Simscape model for the driving subsystem is shown in Figure 10.28.

Ideal torque sensors are connected to the input shaft (LHS) and the output shaft (RHS) to verify that proper relationships between elements have been incorporated into the model. Both sensors are connected to a scope to monitor whether the gearbox acts as a constant speed reducer.

Applying a sine wave input signal (simulating a no-load input speed of 1800 rpm), the ideal gearbox exerts a peak output speed of 16.8 rpm as shown in Figure 10.29. Similarly, with no load and the same input sine wave, the peak torque increases from 0.1584 to 16.88 Nm.

The "driven" subsystem, which consists of a wheel and axle mechanism coupled to a three-linkage mechanism, converts the rotational motion of the AC motor into the translational push–pull movement of the conveying system as shown in Figure 10.30.

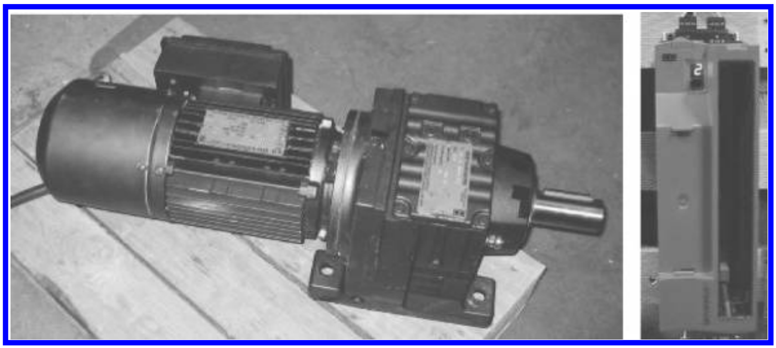

(a) (b)

FIGURE 10.27
Driving subsystem. (a) AC motor and gearbox and (b) VFD.

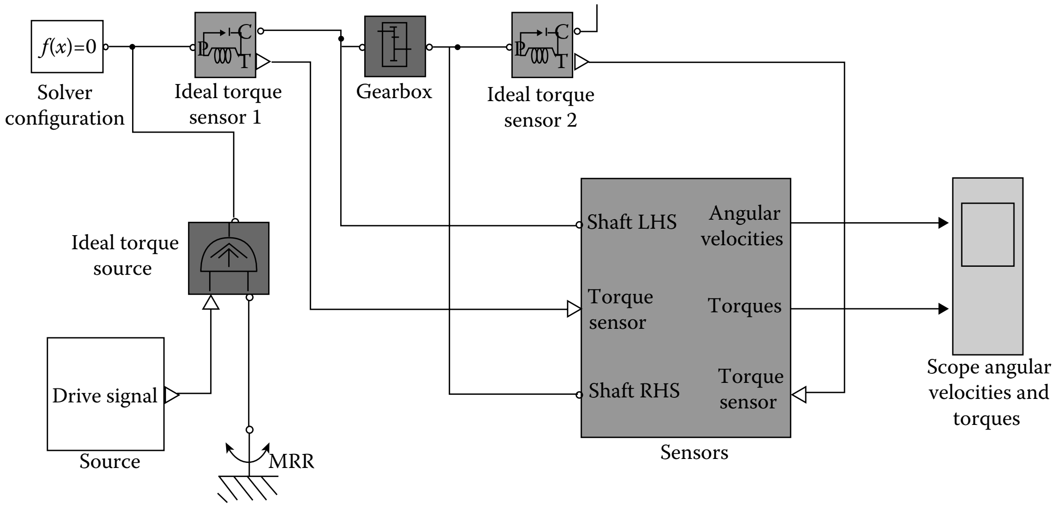

f(x)=0
Solver configuration
Ideal torque sensor 1
Gearbox
Ideal torque sensor 2
Ideal torque source
Drive signal
Source
MRR
Shaft LHS
Angular velocities
Torque sensor
Torques
Shaft RHS
Torque sensor
Sensors
Scope angular velocities and torques

FIGURE 10.28
Simscape model of the driving section.

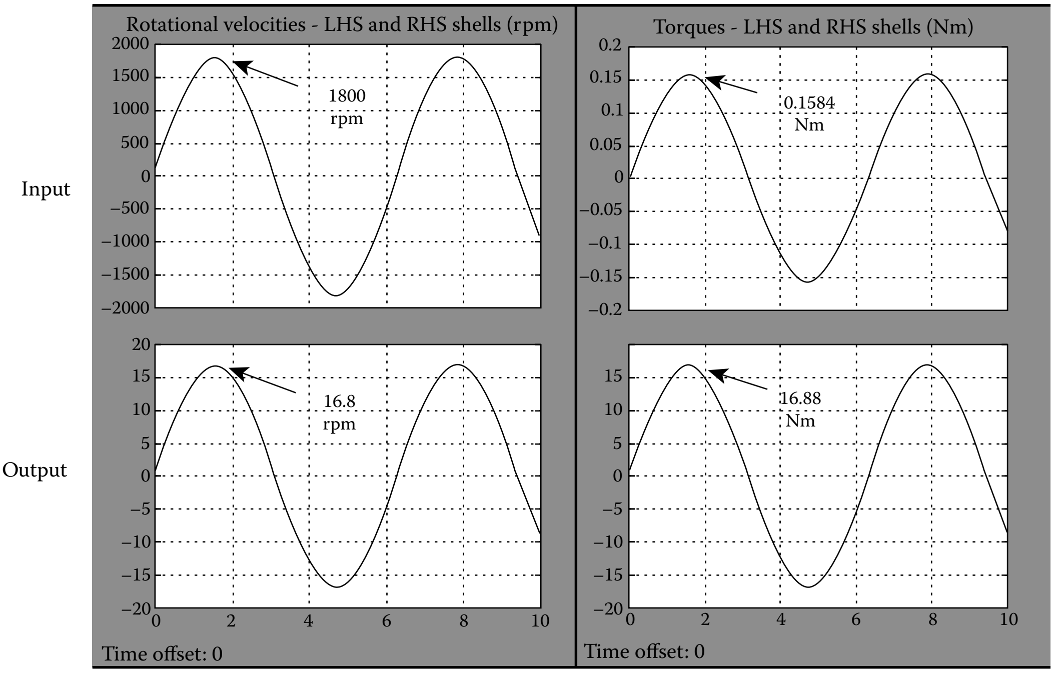


FIGURE 10.29
Gearbox input and output signals.

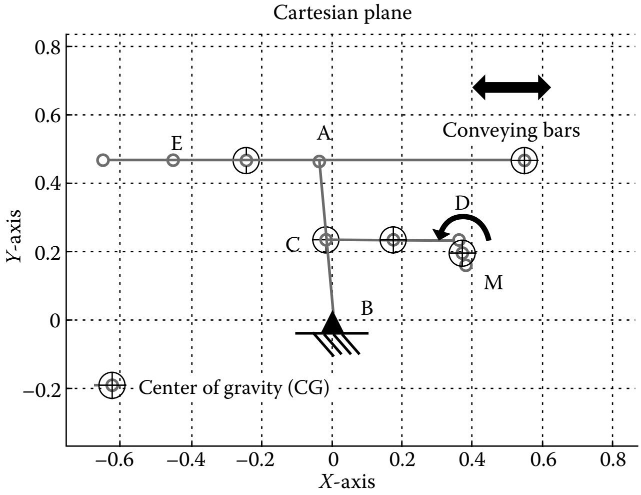


FIGURE 10.30
Illustration of the push–pull movement.

The linkage AE is the final element of the mechanism that transmits the force required to move the sliding copper bars. These copper bars have several indexing pins, which fold only in one direction, thereby allowing the fish to be transported with a linear intermittent movement through the table of the IIB. The model for this section is shown in Figure 10.31. The linkages are represented by blocks called "bodies" (shaded), which include information such as mass, moment of inertia tensor, the location of the center of gravity, and the physical location with respect to the global or local coordinates in the system. The bodies are conveying bars, linkage AE, linkage AB, linkage CD, and disc DM.

Every block is connected to a joint block (revolute joints 1, 2, 3, 4, and 5 and a prismatic joint), which represents the degrees of freedom (DOF) of the body relative to another body or relative to the global reference in the system. Similar to the Simscape model, the SimMechanics model has the possibility of incorporating joint sensors and scopes to visualize responses to different stimuli. After determining the parameters of every component in the system and specifying them in the model along with their units, the simulation has shown the following results for a constant input speed of 1800 rpm.

- Simulation time: 20 sec
- Average distance traveled by the indexing pins per cycle: 30.3 cm
- Average time per cycle: 2.53 sec

Although the calculated parameters are realistic, several assumptions have been made. To verify the correctness of the model, the results are compared with experimental data. For example, the measured distance of the indexing pins traveled on the IIB is approximately 28.5 cm. Also the average time to complete a cycle is 3.05 sec. The existence of differences is understandable as the model has not taken into consideration several aspects, such as the losses due to friction between the copper bars and the support on which they rest.

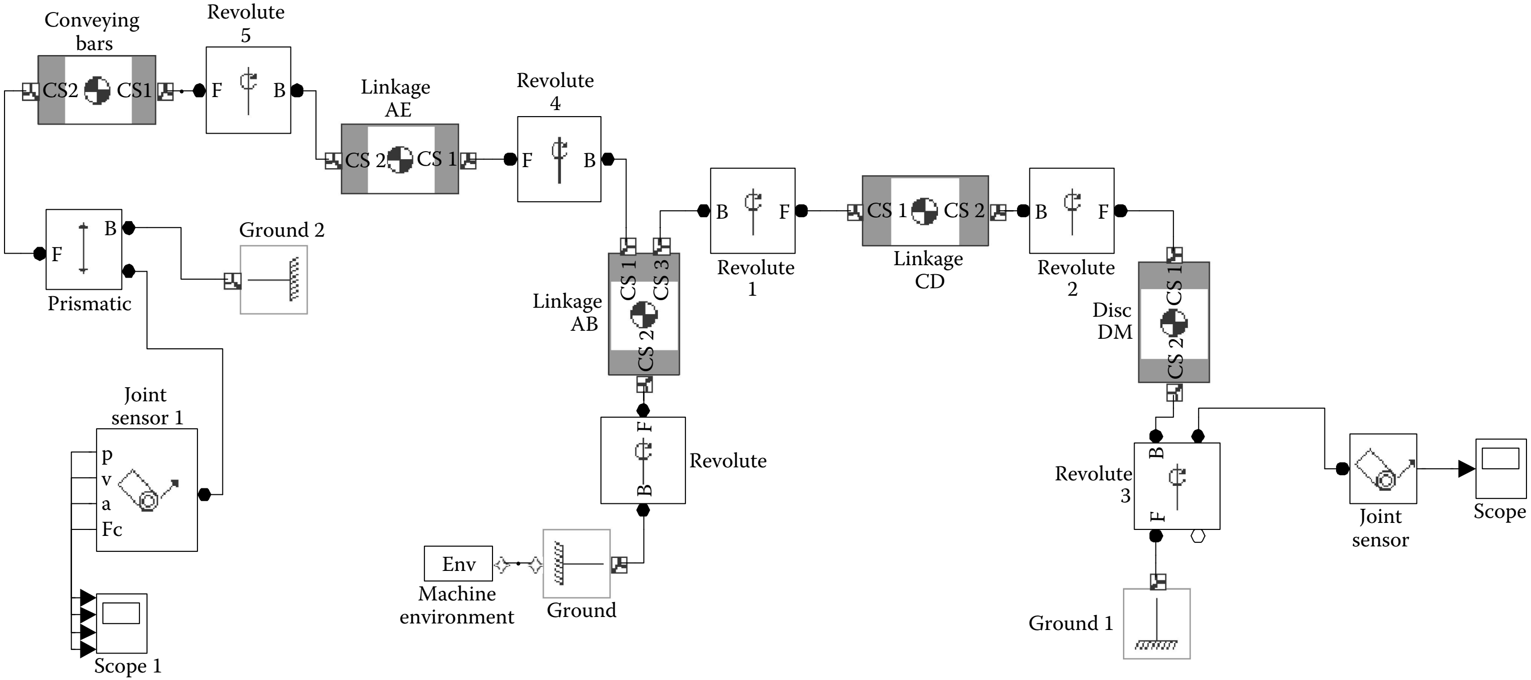

Conveying bars
CS2
CS1
Revolute 5
F
B
Linkage AE
CS 2
CS 1
Revolute 4
F
B
Prismatic
F
B
Ground 2
Linkage AB
CS 1
CS 3
CS 2
B
Revolute 1
B
F
Linkage CD
CS 1
CS 2
Revolute 2
B
F
Disc DM
CS 1
CS 2
Revolute
F
B
Joint sensor 1
p
v
a
Fc
Scope 1
Env
Machine environment
Ground
Revolute 3
B
F
Joint sensor
Scope
Ground 1

FIGURE 10.31
SimMechanics model of the driven section.

10.4.3 Illustrative Scenario 2

The machine HMS has detected that the quality and the accuracy of the fish head removal process have degraded as a result of the conveyor speed being slow [21,22]. The ES has determined, based on the knowledge gathered over a period of time, that the corresponding shortcoming is in the gearbox subsystem due to a design weakness. Therefore, the design objective is to find the proper gearbox with the correct gearbox ratio that yields the required conveyor speed. The design evolution process described above must be employed to find an optimum design solution for the conveying system. As the design weakness has been attributed to the gearbox subsystem, the embryo model developed for evolution must contain modifiable sites in the gearbox region of the model. First, let us understand the process of LG evolution with GP.

10.4.3.1 LG Evolution with GP

As in the case of BGs, a set of construction functions and terminals has to be developed to evolve an LG using GP. A construction function adds a component to a modifiable node of the evolving model and assigns a value to it whereas a terminal ends further evolution of an LG, thereby controlling the arbitrary evolution. Figure 10.32 illustrates how the Add function inserts an element into a modifiable site (node) and creates two new modifiable sites. The add function maintains the modifiable site (a) for further possible modification at the node. It creates a new node at the modifiable site (b) and will add a single-port across-, through-, or damping-type element at the modifiable site (c). The modifiable site (c) allows the branch to change its corresponding parameter value. Table 10.6 shows the functions and terminals used for the evolution.

Note that the *Add_A* function has a special meaning in the mechanical domain. When it is a mass or a moment of inertia element, it has to be connected to a reference point or the ground.

The evolution of a mechatronic system by using the functions and terminals given in Table 10.6 can be illustrated as follows: Consider the embryo LG model shown in Figure 10.33. Evolution is allowed only from modifiable sites marked (a), (b), and (c). These sites are identified by the ES or domain experts with the help of data obtained from the HMS. As in the BG evolution, restricting modifiable sites only to relevant areas of the LG with the help of the available knowledge will guide design exploration avoiding arbitrary evolution.

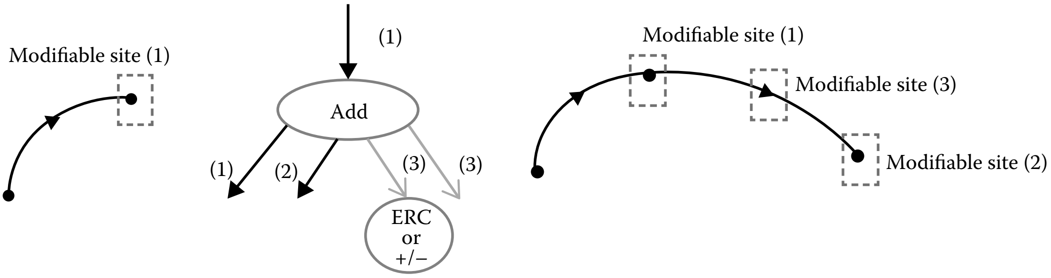


FIGURE 10.32
Illustration of adding an element to a modifiable node.

TABLE 10.6

LG-GP Functions and Terminals

Name	Arity	Description
Add_A	1	Adds an across-type element (mass, moment of inertia, capacitor, thermal capacitor, or fluid capacitor)
Add_T	1	Adds a through-type element (linear spring, torsional spring, inductor, or fluid inerter)
Add_D	1	Adds a damping-type element (viscous damper, resistor, thermal resistor, or fluid resistor)
Add_t	1	Adds a transformer-type element (two-port)
Add_g	1	Adds a gyrator-type element (two-port)
Add_L	2	Adds a line between two modifiable nodes
Replace_A	2	Replaces the current element with an across-type element
Replace_T	2	Replaces the current element with a through-type element
Replace_D	2	Replaces the current element with a damping-type element
Replace_t	2	Replaces the current two-port element with a transformer-type element
Replace_g	2	Replaces the current two-port element with a gyrator-type element
Desprog_2	0	Executes two functions sequentially
Desprog_3	0	Executes three functions sequentially
Close_node	1	Terminates a modifiable node
Gnd	1	Connects a modifiable node to ground reference and terminates it
ERC	0	Ephemeral random constant
+, −, *	2	Adds, subtracts, or multiplies two ERCs
End	0	Terminates a modifiable branch

Figure 10.34 illustrates how the embryo shown in Figure 10.33 is evolved to produce the LG model shown in Figure 10.35. The evolution process starts with an *Add_T* function, which adds a “through-type” element to node (a), computes its value using ERC, and closes further evolution by *Close_Node* operation. The process moves to the other nodes and continues until no further modifiable sites are available. The functions *desprog* and *Add_L* are used to connect nodes to obtain a meaningful (complete) LG. LGs that are not complete after evolution are discarded during the evaluation process.

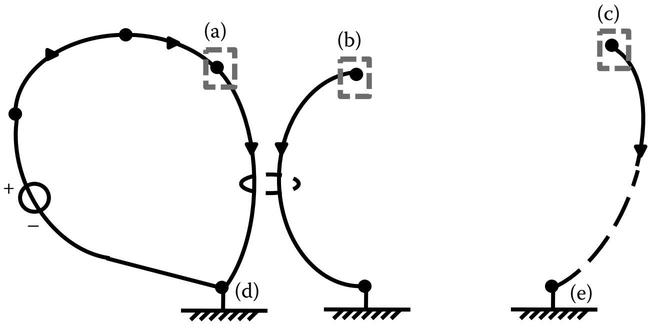


FIGURE 10.33
Embryo example of a mechatronic LG model.

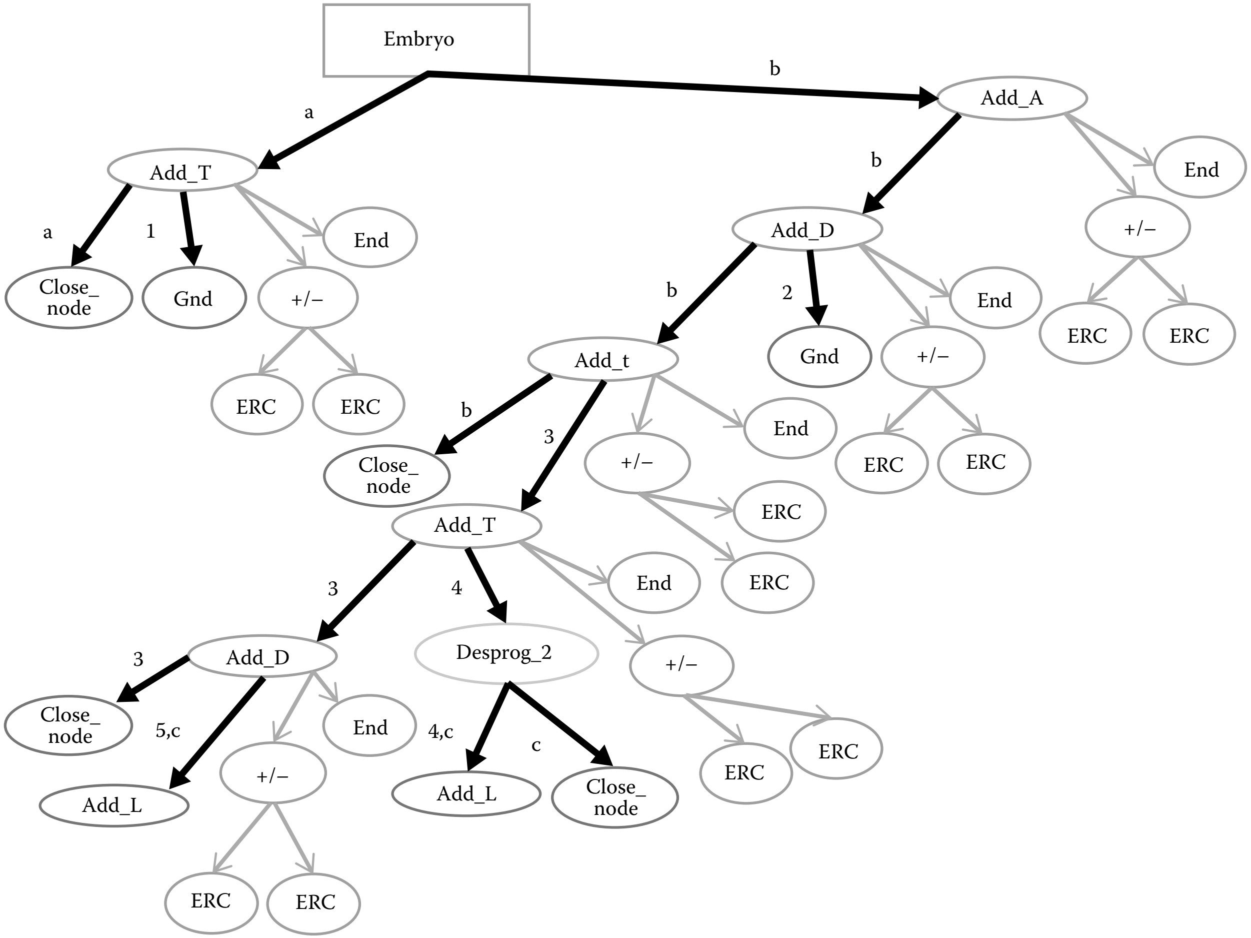

Embryo
a
b
Add_T
Add_A
a
1
Close_ node
Gnd
+/−
End
ERC
ERC
b
Add_D
End
+/−
ERC
ERC
b
2
Add_t
Gnd
+/−
End
ERC
ERC
b
3
Close_ node
+/−
End
ERC
ERC
Add_T
3
4
End
ERC
3
Add_D
Desprog_2
+/−
Close_ node
5,c
End
4,c
c
ERC
ERC
Add_L
+/−
Add_L
Close_ node
ERC
ERC

FIGURE 10.34
Illustration of LG growth from an embryo model.

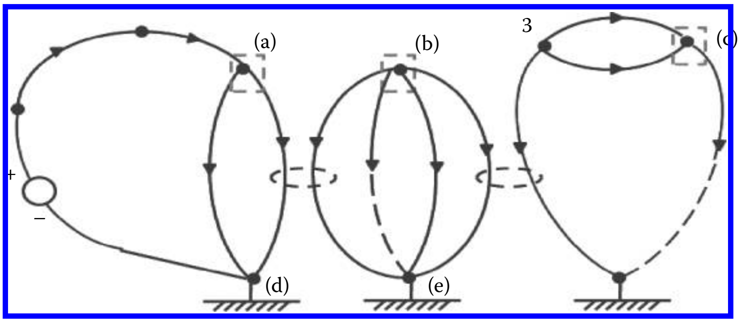


FIGURE 10.35
Evolved LG model by the GP tree.

The algorithm of the design evolution framework is shown in Figure 10.36. The dark shaded blocks identify steps at which the HMS and the ES exchange information so that the evolutionary process can take place. Operations executed by the evolutionary process, such as LG model creation, embryo model initialization with modifiable sites, alternatives evolution, and simulation of feasible design candidates, are indicated in light shaded blocks.

10.4.3.2 Design Improvement of the Conveying System

As the weakness has been identified in the gearbox subsystem, the components corresponding to the gearbox are removed from the LG model, and two modifiable sites are introduced as shown in Figure 10.37 in place of these components to commence the evolution process.

The developed embryo model was converted into a Simscape model for the purpose of evolution using GP. The performance of each evolved individual through GP was compared with a specified fitness function, which corresponds to a design specification of 30 cm displacement of indexing pins in 2.5 sec. The input is kept at constant speed of 1800 rpm. In addition to comparing the performance fitness, factors such as complexity of the evolved individual and the variation of component parameters from the current values were also considered.

The evolution was carried out for 50 generations with a population size of 100. The best GP tree individual was found in generation 47, and it had a fitness value of 0.0300. The best GP tree corresponds to a design of conveying system with a gearbox ratio of 106.58:1 compared to the installed gearbox with a ratio of 147.92:1. The installed gearbox had an average time per cycle of 3.04 sec with an average distance traveled by the indexing pins of 30.5 cm. The operators of the IIB would have had to increase the motor speed to compensate for the 0.54-sec delay in every cycle, and as a result, the gearbox would have been functioning outside its design specifications causing it to fail.

The new gearbox with the ratio of 106.58:1 produced an average time per cycle of 2.53 sec, which satisfies the original design specification of 2.5 sec per cycle.

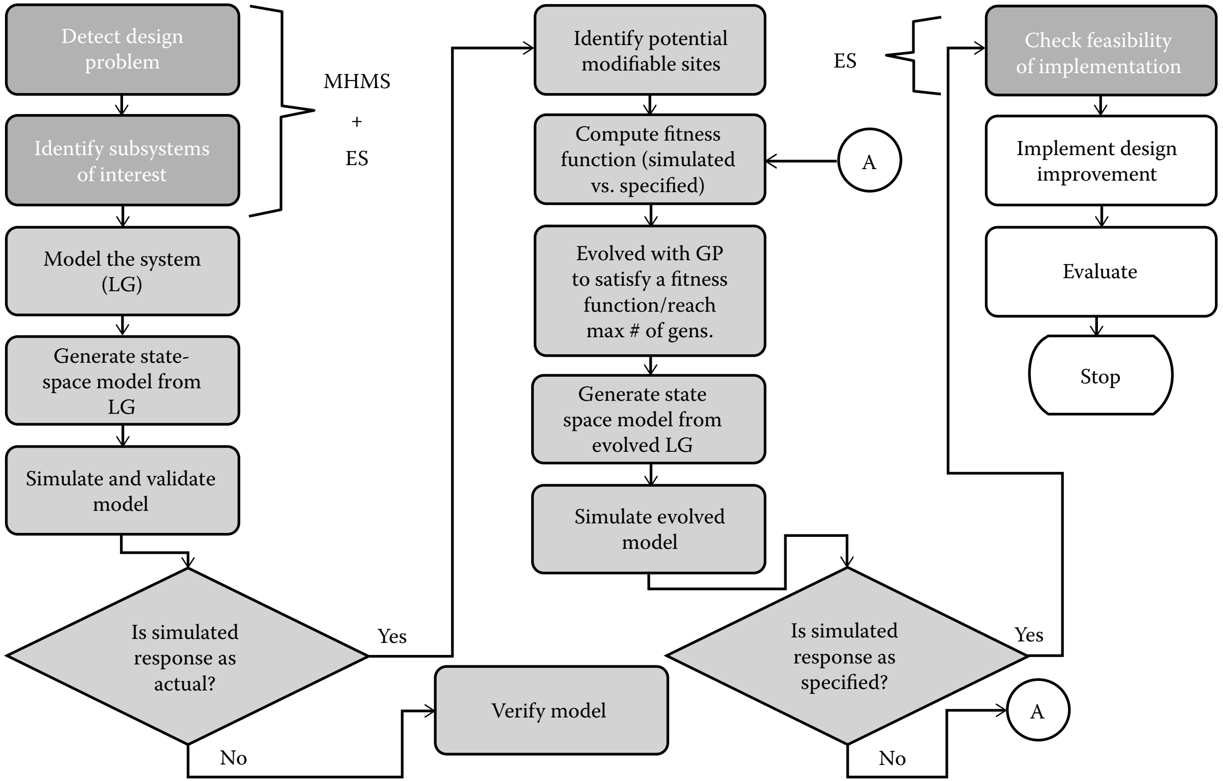

Detect design problem
Identify subsystems of interest
MHMS
+
ES
Model the system (LG)
Generate state-space model from LG
Simulate and validate model
Is simulated response as actual?
Yes
No
Verify model
Identify potential modifiable sites
Compute fitness function (simulated vs. specified)
A
Evolved with GP to satisfy a fitness function/reach max # of gens.
Generate state space model from evolved LG
Simulate evolved model
Is simulated response as specified?
Yes
No
A
ES
Check feasibility of implementation
Implement design improvement
Evaluate
Stop

FIGURE 10.36
Algorithm of the design evolution framework.

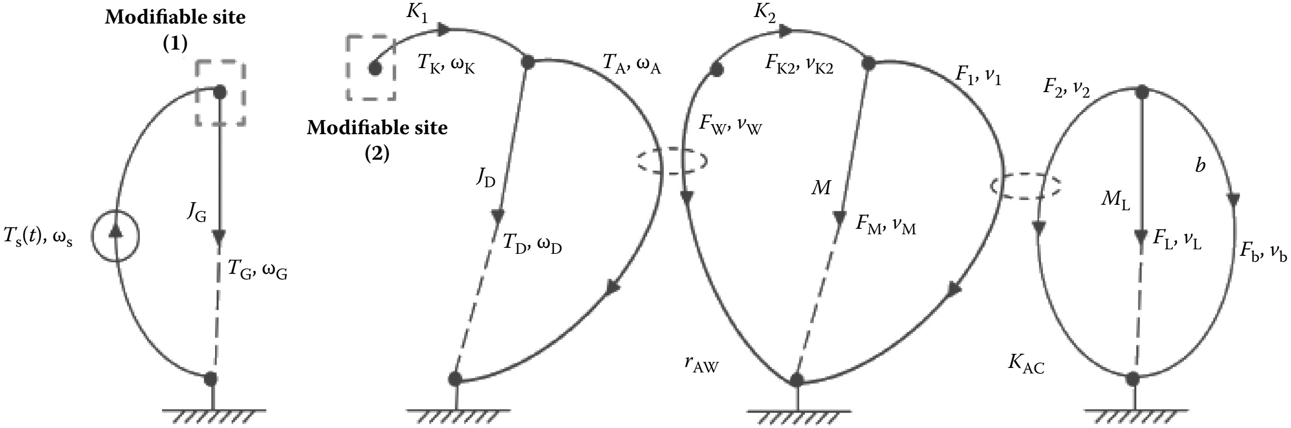

Modifiable site
(1)
$T_s(t)$, ω_s
J_G
T_G, ω_G
K_1
Modifiable site
(2)
T_K, ω_K
T_A, ω_A
J_D
T_D, ω_D
K_2
F_{K2}, v_{K2}
F_1, v_1
F_W, v_W
M
F_M, v_M
r_{AW}
F_2, v_2
b
M_L
F_L, v_L
F_b, v_b
K_{AC}

FIGURE 10.37
LG embryo model with two modifiable sites.

10.5 Conclusions

This chapter presented a system framework for design evolution of multidomain engineering systems or mechatronic systems. The described framework integrated machine health monitoring with an expert system to monitor the performance of an existing mechatronic system and to make a decision as to whether a design improvement is necessary. If a design improvement is necessary, the ES will also assist in identifying the section of the system that needs to be improved. Modeling of multidomain engineering systems using BGs and LGs was explained in detail with examples. A technique to evolve a system model represented by a BG or a LG using GP was introduced. This technique was combined with a fitness function that represented a set of desired performance specifications to obtain an optimum design. The developed approach allowed exploration of the design space to find the optimum design solution in an automated manner. In order to control the arbitrary exploration of the design space, domain knowledge, expertise, and input from the machine HMS were used. The design evolution algorithm was implemented using GPLAB, a MATLAB tool, and integrated with Simscape for modeling and simulation. The developed system was applied to an industrial fish-processing machine to demonstrate its capability in addressing performance issues by redesigning two subsystems of the machine. Also, to illustrate the usability of the two modeling techniques, BG modeling and LG modeling were employed to redesign the hydraulic cutter positioning system and electromechanical conveyor system of the fish-processing machine, respectively.

References

1. C. W. de Silva, *Mechatronics—A Foundation Course*, CRC Press, Taylor & Francis, Boca Raton, FL, 2010.
2. C. W. de Silva, *Modeling and Control of Engineering Systems*, CRC Press, Taylor & Francis, Boca Raton, FL, 2009.
3. C. W. de Silva, *Mechatronics—An Integrated Approach*, CRC Press, Taylor & Francis, Boca Raton, FL, 2005.
4. S. Raman and C. W. de Silva, "Classifier Design for Sensor-Fault Tolerant Condition Monitoring in an Industrial Machine," *Proc. ASME Dynamic Systems and Control Conference*, Hollywood, CA, pp. 1–7, 2009.
5. C. W. de Silva, "A Unified System Framework Including Intelligent Supervisory Control and On-Line Design Evolution with Networked Remote Operation," *A Proposal for a Tier 1 Canada Research Chair*, Natural Sciences and Engineering Research Council, Ottawa, Canada, 2008.
6. L. B. Gamage and C. W. de Silva, "A System Framework with On-line Monitoring and Evaluation for Design Evolution of Engineering Systems," *Journal of Computing and Information Science in Engineering*, Vol. 10, pp. 1–6, 2010.
7. L. B. Gamage, C. W. de Silva, and R. Campos, "Design Evolution of Mechatronic Systems through Modeling, On-line Monitoring, and Evolutionary Optimization," *Mechatronics*, Vol. 22, No. 1, pp. 83–94, 2012.
8. J. Hu, E. Goodman, and R. Rosenberg, "Topological Search in Automated Mechatronic System Synthesis Using Bond Graphs and Genetic Programming," *Proc. American Control Conference*, Boston, pp. 5628–5634, 2004.

9. C. W. de Silva and S. Pourazadi, "Some Generalizations of Linear-graph Modeling for Dynamic Systems," *International Journal of Control*, Vol. 86, No. 11, pp. 1990–2005, 2013.
10. C. W. de Silva, "Linear-graph Modeling Paradigms for Mechatronic Systems," *The International Journal of Mechanical Engineering Education*, Vol. 40, No. 4, pp. 305–328, 2012.
11. K. Seo, Z. Fan, J. Hu, E. D. Goodman, and R. C. Rosenberg, "Toward a Unified and Automated Design Methodology for Multi Domain Dynamic Systems Using Bond Graphs and Genetic Programming," *Mechatronics*, Vol. 13, pp. 851–885, 2003.
12. J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, "Synthesis of Topology and Sizing of Analog Electrical Circuits by Means of Genetic Programming," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, pp. 459–482, 2000.
13. S. Behbahani and C. W. de Silva, "Mechatronic Design Quotient as the Basis of a New Multicriteria Mechatronic Design Methodology," *IEEE-ASME Transactions on Mechatronics*, Vol. 12, pp. 227–232, 2007.
14. S. Behbahani and C. W. de Silva, "Mechatronic Modeling and Design," *Mechatronic Systems—Devices, Design, Control, Operation, and Monitoring*, C. W. de Silva (ed.). CRC Press, Taylor & Francis, Boca Raton, FL, 2007.
15. S. Behbahani and C. W. de Silva, "Niching Genetic Scheme with Bond Graphs for Topology and Parameter Optimization of a Mechatronic System," *IEEE/ASME Transactions on Mechatronics*, Vol. 19, No. 1, pp. 269–277, 2014.
16. C. W. de Silva and S. Behbahani, "A Design Paradigm for Mechatronic Systems," *Mechatronics*, Vol. 23, No. 8, pp. 960–966, 2013.
17. F. Karray and C. W. de Silva, *Soft Computing and Intelligent Systems Design*, Addison-Wesley, New York, 2004.
18. C. W. de Silva, "Sensing and Information Acquisition for Intelligent Mechatronic Systems," *Science and Technology of Information Acquisition and Their Applications, Proceedings of the Symposium on Information Acquisition*, Chinese Academy of Sciences, Hefei, China, pp. 9–18, November 2003 (Invited, Keynote Address).
19. C. W. de Silva, "Sensory Information Acquisition for Monitoring and Control of Intelligent Mechatronic Systems," *International Journal of Information Acquisition*, Vol. 1, No. 1, pp. 89–99, 2004.
20. C. W. de Silva, "Research Laboratory for Fish Processing Automation," *International Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 9, No. 1, pp. 49–60, 1992.
21. H. Lang and C. W. de Silva, "Fault Diagnosis of an Industrial Machine through Sensor Fusion," *International Journal of Information Acquisition*, Vol. 5, pp. 93–110, 2008.
22. B. Razavi and C. W. de Silva, "Condition Monitoring in a Hydraulic System of an Industrial Machine Using Unscented Kalman Filter," *International Journal of Information Acquisition*, Vol. 7, pp. 177–192, 2010.

11

Mechatronic Design of Unmanned Aircraft Systems

Feng Lin, Fei Wang, Xiangxu Dong, Kemao Peng, and Ben M. Chen

CONTENTS

11.1 Introduction 404
11.2 Unmanned System Hardware 407
  11.2.1 Sensors and Measurement Systems 407
    11.2.1.1 Inertial Sensors 407
    11.2.1.2 GPS/DGPS 410
    11.2.1.3 Magnetometer 410
    11.2.1.4 Lidar 411
    11.2.1.5 Vision Sensor 412
    11.2.1.6 RGB-D Camera 413
  11.2.2 Computers 415
  11.2.3 Actuator Management 415
  11.2.4 Communication Unit 416
  11.2.5 Hardware Integration 417
11.3 Unmanned System Software 421
  11.3.1 Onboard Real-Time Software System 421
  11.3.2 Ground Control Software System 424
11.4 Case I: Design of a Coaxial Rotorcraft System 424
  11.4.1 Hardware System 424
    11.4.1.1 Navigation Sensors 427
    11.4.1.2 Computers 427
    11.4.1.3 Servo Controller 428
    11.4.1.4 Communication 429
    11.4.1.5 Control Hub 429
    11.4.1.6 Hardware Integration 429
  11.4.2 Software System 432
    11.4.2.1 Onboard Real-Time Software System 432
    11.4.2.2 Ground Control Software System 433
  11.4.3 Experimental Results 434
11.5 Case II: Design of a UAV Cargo Transportation System 436
  11.5.1 Hardware System 436
    11.5.1.1 Grabbing Mechanism 438
    11.5.1.2 Sensors and Measurement Systems 440
    11.5.1.3 Computers 441
  11.5.2 Software System 442
  11.5.3 Experimental Results 445
11.6 Conclusion 448
References 448

SUMMARY This chapter presents the mechatronics design of unmanned aircraft systems to deliver autonomous capabilities in various missions. The modular design approach and computer-aided hardware integration have been addressed to achieve cost-effective development of the hardware system. The multitask embedded software design has been presented in the software system development to tackle complex missions. The proposed design approaches are feasible and effective, which have been verified in two case studies including a coaxial rotorcraft GremLion developed for the DARPA UAVForge challenge and a cargo transportation system used in UAVGP 2013. Such design approaches can also be extended to different unmanned systems for autonomous missions.

11.1 Introduction

Mechatronics is the synergistic combination of mechanical engineering, electrical engineering, control engineering, and computing engineering and also systems thinking in the design and development of products and manufacturing processes. An unmanned aerial vehicle (UAV) is a typical mechatronics system (see Figure 11.1), which is equipped with necessary data processing units, advanced sensors, an actuation system, automatic

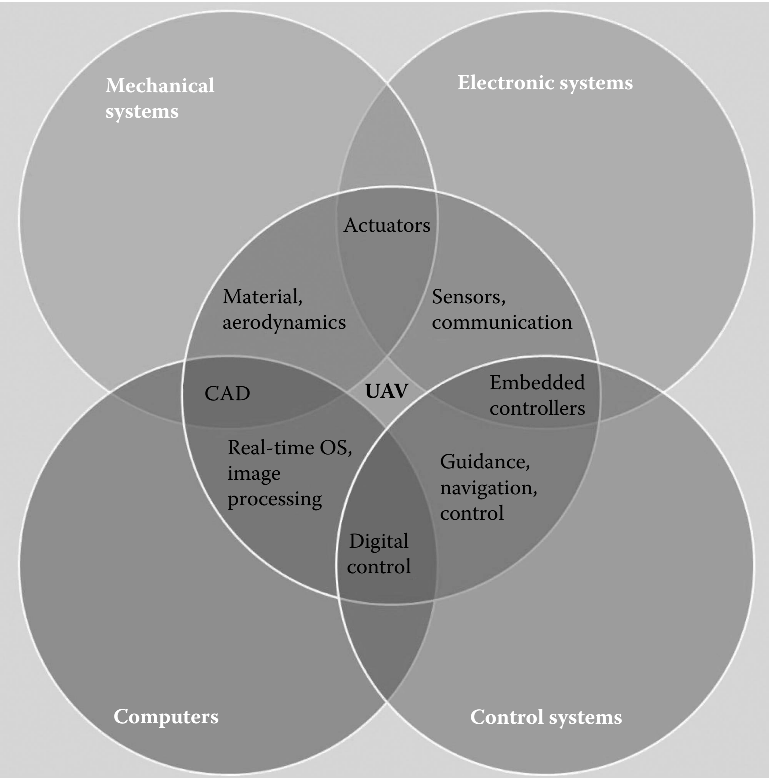

Mechanical systems
Electronic systems
Actuators
Material, aerodynamics
Sensors, communication
CAD
UAV
Embedded controllers
Real-time OS, image processing
Guidance, navigation, control
Digital control
Computers
Control systems

FIGURE 11.1
Mechatronics integration of UAVs.

control, and communication systems in order to perform autonomous flight missions without an onboard crew [1]. To fit its systematic nature, a UAV system is designed and developed in an integrated, cross-disciplinary manner.

In the last two decades, UAVs have made explosive growth in the civil and industrial markets, ranging from surveillance, reconnaissance, and agriculture to wildlife conservation [2–7]. It can be foreseen that the demand for UAVs will continue to grow, and they will become a valuable tool for researchers, rescuers, and other users in various working areas [8–15].

For instance, UAVs have been adopted as useful platforms to verify advanced flight algorithms. In [16], the researchers have proposed a hierarchical nonlinear flight controller that combines nonlinear feedback control for the inner loop and dynamic inversion for the outer loop. The proposed controller has been verified in a real flight test using a Raptor 90 helicopter equipped with the customized avionics to achieve automatic takeoff and landing, hovering, slithering, pirouetting, vertical turning, and spiral turning. In [17], a bumpless hybrid supervisory control algorithm has been applied to the formation control of UAVs, based on polar abstraction of the motion space and the use of properties of multi-affine functions over the partitioned space. Several actual flights have been conducted to evaluate the proposed algorithms using unmanned helicopters.

Another interesting research direction is to enhance navigation and localization capabilities of UAVs in extreme conditions, such as GPS-denied unknown environments. Simultaneous localization and mapping (SLAM) have been studied extensively by fusing the measurements of onboard sensors only [18–21]. Especially, vision-aided SLAM solutions have been tested successfully in [22,23], which are extremely useful for ultralight UAVs, such as insect-size flapping-wing UAVs.

In many practical applications, low-level autonomy has already been achieved successfully, such as obstacle avoidance, path planning, and trajectory tracking. Due to limitations in advanced perception and artificial intelligence, research in high-level autonomy of single or multiple UAVs in unsupervised situations is still in the preliminary stage and definitely needs great efforts to make a breakthrough. For instance, a formal framework and architecture using delegation has been proposed in [24] to support the specification, generation, and execution of a collaborative mission for multiple UAVs and associated operators. The cooperative search and exploration using multiple UAVs has been addressed in [25]. Moreover, there is a trend to combine UAVs with other unmanned systems, such as unmanned ground vehicles and unmanned surface vehicles, to form a large-scale and effective integrated unmanned system [26].

To satisfy the requirements of the aforementioned exciting applications, UAV systems need to be designed and developed in a systematic way. A typical UAV system consists of the following four parts: an aircraft, avionics, a manual control system, and a ground supporting system.

1. The aircraft is a platform to carry avionics and other payloads, which consist of the structure and the propulsion system.
2. Avionics are the onboard electronic systems that have the functions of sensing, stabilizing, navigating, perceiving, analyzing, communicating, planning, decision-making, and acting or executing. Avionics are composed of sensing and measurement systems, computers, communication units, embedded software systems, and associated equipment.
3. The ground supporting system provides a user-friendly interface for the display and management of UAVs. It includes a ground wireless transceiver and a

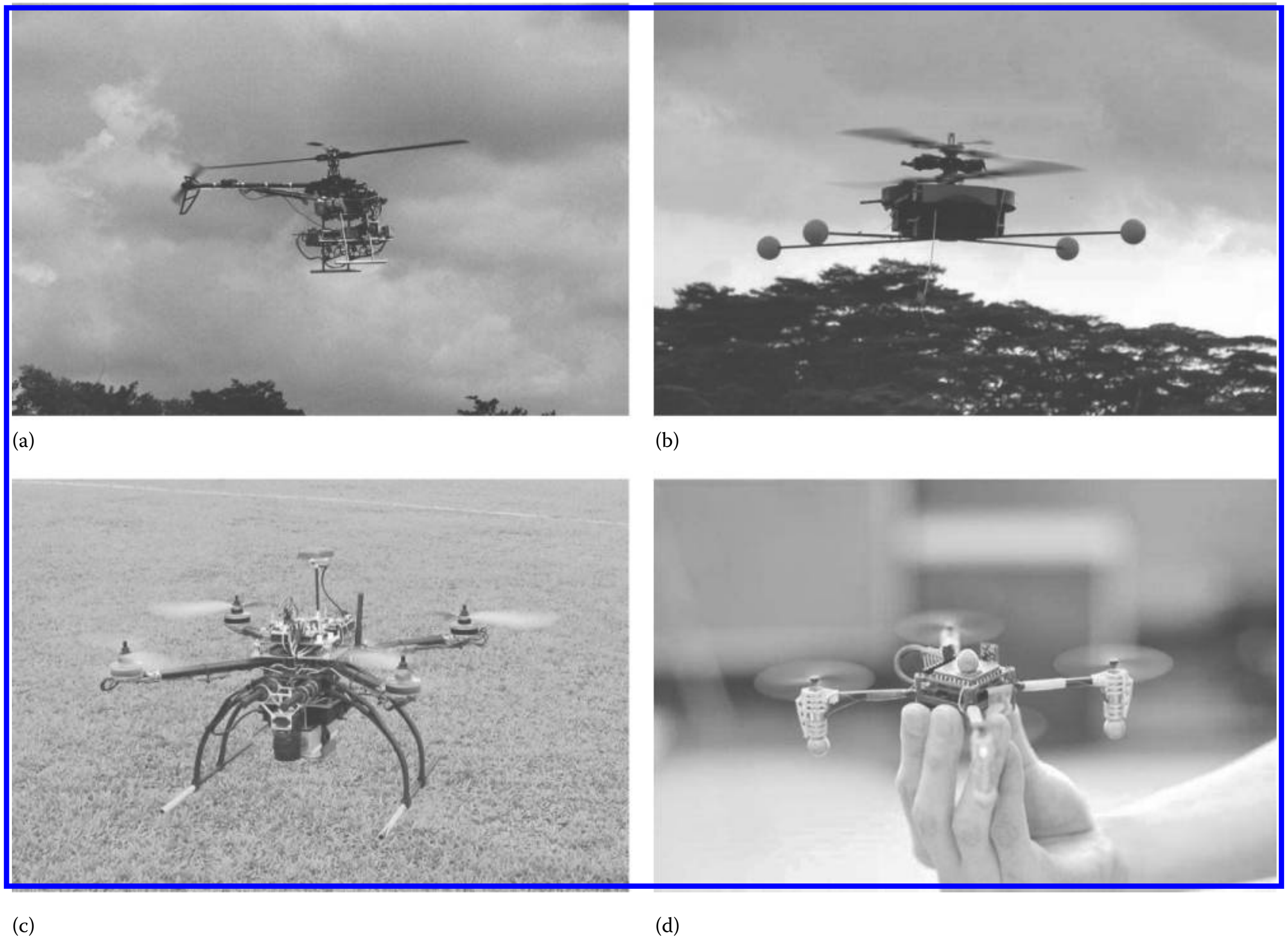


FIGURE 11.2
UAVs developed by the Unmanned System Research Group in NUS: (a) HeLion, (b) GremLion, (c) Q-Lion, and (d) K-Lion.

computer, which is able to monitor the states of the UAV in real time and send the command to the UAV through this interface.

4. The manual control system is a backup system for UAVs, which normally consists of a pilot and a wireless joystick. It is very useful in unpredicted events, such as emergency landing or mechanical or electrical failures of UAVs.

In this chapter, we aim to address systematic design approaches of the avionics and ground control system from the perspective of mechatronics, including hardware and software systems. We will further illustrate the design approaches by using a couple of UAV systems developed by the Unmanned System Research Group in the National University of Singapore (shown in Figure 11.2). Due to word limit, the platform design and the manual control system are not covered in this chapter. Reader please refer to [27,28] for more details. We emphasize that the UAV systems discussed in this chapter belong to the category of mini UAVs [28] and are mainly used for academic and civilian applications.

The remainder of this chapter is organized as follows: The hardware system design and integration is presented in Section 11.2. The software system development is presented in Section 11.3. Several case studies are explored in Sections 11.4 and 11.5. Finally, we draw the conclusion in Section 11.6.

11.2 Unmanned System Hardware

Avionics are composed of various sensors, computers, actuators, and communication units as illustrated in Figure 11.3, and they can be divided into multiple self-contained functional modules and implemented using the modular design approach. Each module needs to be developed or purchased separately with industrial standard interfaces. Such modularity offers benefits such as a reduction in cost and flexibility augmentation. The functions and features of those essential modules and the hardware integration issues will be addressed in the following sections.

11.2.1 Sensors and Measurement Systems

Unlike a radio controlled "drone," a UAV is an autonomous system that is able to sense, make decisions, and react without human interaction. The onboard sensors are combined together to obtain the best-achievable estimation, including 6 DoF rigid body motion and the perception of the surrounding environment. The characteristics of those key sensors are addressed in this section, and the implementation will be presented in Sections 11.4 and 11.5.

11.2.1.1 Inertial Sensors

An inertial navigation system (INS) is a self-contained and core navigation solution, which can dead reckon the attitude, velocity, and position without external navigation references. The basic element of INS is an inertial measurement unit (IMU) composed of a three-axis accelerometer measuring the linear acceleration in the inertial reference frame resolved in the carrier's body coordinate, and a three-axis gyroscope providing the angular rates in the inertial reference coordinate. A couple of examples of INS have been

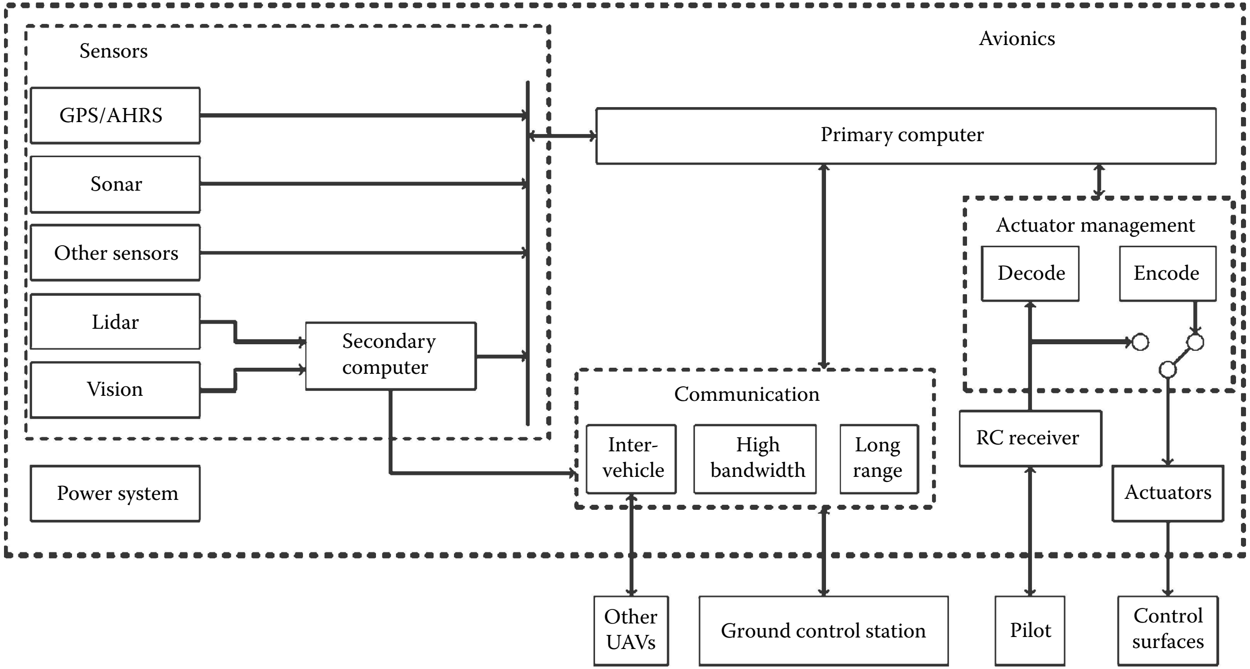

Avionics
Sensors
GPS/AHRS
Sonar
Other sensors
Lidar
Vision
Secondary computer
Power system
Primary computer
Actuator management
Decode
Encode
Communication
Inter-vehicle
High bandwidth
Long range
RC receiver
Actuators
Other UAVs
Ground control station
Pilot
Control surfaces

FIGURE 11.3
Hardware configuration of a UAV.

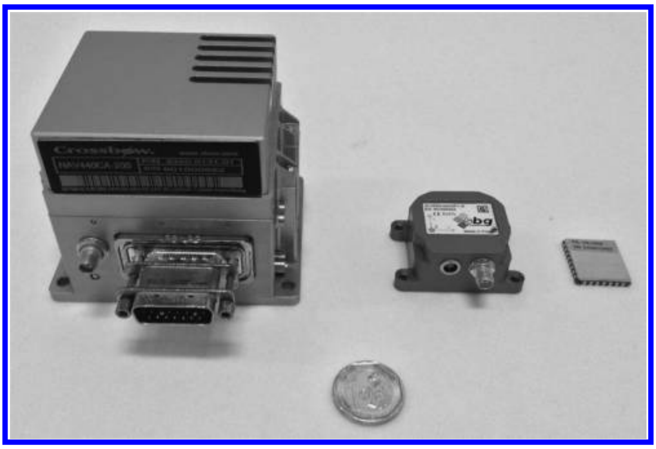


FIGURE 11.4
Inertial navigation systems (left to right): NAV440, IG500N, and Vectornav VN-100.

shown in Figure 11.4. A comprehensive list of commercial available small IMUs and INSs can be found in [29].

The navigation accuracy of the INS is significantly affected by the IMU used, and IMUs can be roughly divided into four performance categories: marine and aviation grade, tactical grade, industrial grade, and automotive and consumer grade [30]. The specification differences of the four performance categories are given in Table 11.1 (see detailed description in [31]), based on the commercial available IMU and INS systems. The bias and noise level of the sensors eventually determine the grade of IMU.

To improve the navigation performance of INSs, especially industrial- or tactical-grade ones, the inertial sensors need to be integrated with a three-axis magnetic sensor to form an attitude and heading reference system (AHRS) that is able to provide filtered attitude and heading solutions with an onboard processing system in the AHRS. In addition to the magnetic sensor, it is common and useful to employ a GPS receiver together with the AHRS to obtain bounded position and velocity estimation. The whole navigation system is referred to as the GPS-aided AHRS or the GPS-aided INS that is able to provide complete navigation information. Moreover, to improve altitude estimation, a barometer is

TABLE 11.1
IMU Categories

IMU Grade	Marine and Aviation	Tactical	Industrial	Automotive
Cost (USD)	100 k–1 M	10–30 k	0.5–3 k	<500
Gyro type	Laser/fiber optic	Fiber optic/MEMS	MEMS	MEMS
Gyro bias (°/h)	0.001–0.01	0.01–1	1–100	>100
Gyro random walk ($°/\sqrt{h}$)	<0.005	0.005–0.5	0.5–5	>5
Accelerometer bias (mg)	0.01–0.1	0.1–1	1–10	>10
Example	Honeywell HG9900 Honeywell HG9848	Honeywell HG1900 Spatial FOG	Crossbow IMU440 Microstrain GX2 SBG IG500A Vectornav VN-100	ArduIMU RasorIMU

integrated into the navigation system. In indoor applications, the external reference system may come from a motion-capture system, providing position and velocity measurements.

There are a large number of small IMU/INS devices available on the market nowadays. To select a suitable one for a UAV system, we are in accordance with the following concerns:

1. If the extreme or acrobatic flight conditions are not taken into consideration, the industrial-grade IMU/INS devices are strongly recommended, which have acceptable performance, affordable prices, and low maintenance cost. For unconventional or fast maneuverable aircrafts, the acceleration, angular rate, and magnetics may change dramatically during flight. As the conventional expression is mathematically singular in one of the flight modes, the quaternion-based motion estimation is compulsory in such situations. In addition, a high sampling rate and large measurement range are also required in such applications.
2. On the basis of meeting all of the abovementioned requirements, the size, weight, and power consumption of the adopted IMU/INS device should be minimized.
3. Pure inertial navigation suffers from integration drift [30]. In practice, inertial navigation needs to be aided by external references, such as GPS, magnetometer, altimeter (barometric, lidar, radar), and stereo vision to realize drift-free state estimation.

11.2.1.2 GPS/DGPS

GPS is widely used nowadays in numerous civil and military applications. The GPS can be used to correct the bias and error of the inertial navigation system. The majority of GPS receivers that are commercially available work at L1-band (1.57542 GHz). Information provided by the GPS receivers commonly includes position in the geodetic coordinate system with about 3 m CEP (3 m circular error probable [32]), velocity in the vehicle-carried NED frame (with the accuracy about 0.5 m/s), and time when the information package is sent.

The accuracy of position and velocity can be further improved to the meter or even centimeter level by employing advanced positioning methodologies, such as differential GPS (DGPS) and real-time kinematic (RTK) navigation methods. However, a stationary GPS base station is required in order to achieve high accuracy measurement. GPS measurement is drift-free and thus commonly blended into the estimation algorithm as the periodical reference signal. Two major deficiencies of GPS receivers are the vulnerability in a poor-visibility environment and low update rate (commonly 1 to 4 Hz).

11.2.1.3 Magnetometer

A magnetometer is to measure the magnetic fields of the Earth. A three-axis magnetometer can be used for estimating heading and gyro bias. In UAV systems, the magnetometer can effectively provide initial reference and periodical correction for the heading angle via measuring the strength of the magnetic field. Most of the commercial magnetometers are MEMS-based and with a sufficient resolution (milligauss level) and sampling rate (up to 10 Hz). For specific implementation in miniature UAVs, special attention should be paid to electromagnetic interference (EMI) shielding and hard- and soft-iron calibrations for each particular platform.

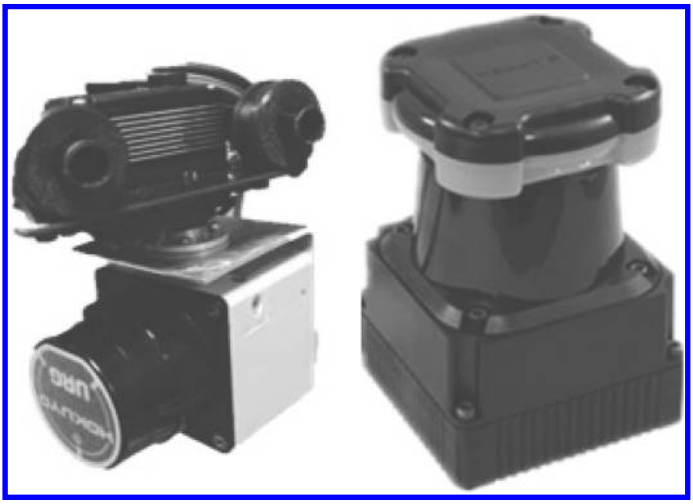
11.2.1.4 Lidar

Relative sensing in unknown environments is a challenging and exciting research topic for autonomy of unmanned systems. The most reliable and promising solutions for small-scale UAVs are lidars and stereo vision. We will introduce lidars first, and the detailed description of stereo vision will be addressed later.

The lidar is a laser-based sensing technology measuring distance by computing time of flight of the laser beam. The lidar technology has shown its efficiency, accuracy, and reliability in many applications, such as surveying, mapping, and autonomous navigation. As an active sensing technique, lidar systems can work under a wide range of environmental conditions, such as day, night, dusk, and dawn. Lidar systems have been used for several decades, but the previous lidar systems were so heavy and bulky that they could only be used in manned aircrafts [33]. With the rapid growth of small-size unmanned systems, there are a variety of ultralight lidar systems available now (for example, see Figure 11.5). According to the type of the scanned area, lidar systems can be categorized into two types: 2-D and 3-D lidars.

A typical 2-D lidar is composed of a laser range finder and a spinning mirror that can directly provide accurate displacement measurements in the scanning plan. Compared to 3-D lidars, 2-D lidars are lightweight and have low power consumption and relatively low cost, which makes them suited for unmanned systems navigating in structured environments [19]. A comparison of commercial off-the-shelf 2-D lidar products is given in Table 11.2. However, the quantity of information encapsulated in a 2-D lidar is not sufficient for crucial and demanding tasks such as obstacle detection in cluttered environments where complete 3-D information is required.

A 3-D scanning lidar obviously can provide much more information about the environment in just one scan. One straightforward way to generate 3-D measurement is to install a 2-D lidar unit in a sweeping or spinning mechanism. Another sophisticated approach is to use the configuration of an array of individual laser sensors, and the entire system



(a) (b)

FIGURE 11.5
2-D lidars: (a) URG-04LX and (b) UTM-30LX.

TABLE 11.2

2-D Lidar Categories

2-D Lidar	Hokuyo URG-04LX	Hokuyo UTM-30LX	SICK LMS511	RIEGL VQ-180
Detection range (m)	4	30	80	150
Accuracy (mm)	10	30	50	15
Scan angle (°)	240	270	190	100
Angular resolution (°)	0.36	0.25	0.25	0.001
Scan speed (ms)	100	25	13	100
Power consumption (W)	3	8	22	50
Weight (kg)	0.16	0.21	3.7	9

TABLE 11.3

3-D Lidar Categories

3-D Lidar	Velodyne HDL-32	Frauhofer 3DLS	Riegl LMS-Z420i	Ocular Robotics RE05
Detection range (m)	70	30	1000	30
Scan angle, vertical (°)	40	124	124	70
Scan angle, horizontal (°)	360	180	360	360
Angular resolution, vertical (°)	0.16	0.25	0.002	0.01
Angular resolution, horizontal (°)	1.33	0.25	0.0025	0.002
Scan speed, vertical (sec)	0.1	26.64	0.05	0.05
Scan speed, horizontal (sec)	0.1	26.64	24	0.05
Data rate (points/sec)	7×10^5	7.6×10^3	1.1×10^4	3×10^4
Power consumption (W)	12	20	78	70
Weight (kg)	2	7.4	16	2.5

spins. The most popular and successful 3-D lidar is the Velodyne lidar, which has been widely used in the DARPA urban challenge. Recently, a Velodyne lidar has been used on a quadrotor for 3-D mapping and surveillance [34]. A comparison of commercial off-the-shelf 3-D lidars is given in Table 11.3. A more detailed comparison and discussion of 3-D lidars is given in [35], which also provides the 3-D coverage of different lidar systems.

The main drawback of laser sensing is the weight and high power consumption due to the nature of active sensing. Although a 3-D lidar is able to provide one more dimensional measurement than a 2-D lidar, it is hard for it to be carried by a small-scale UAV with limited payload and power supply. Therefore, in localization and navigation applications, the combination of two 2-D lidars is a feasible solution, balancing the quality of 3-D measurement against total weight. In addition, fusing a lidar with other sensors, such as a camera, can also provide instant 3-D information [36]. An application of using a lidar system on a small-scale UAV will be presented in Section 11.5. In practice, measurements of the lidar are affected by the attitude and velocity of the carrying vehicles, and it is necessary to do motion compensation for a moving platform.

11.2.1.5 Vision Sensor

There is a boom in vision sensing in academic research and industrial applications because it is able to provide human-like perception, such as geometry of the scene, photometry of objects, and dynamics of the environment. By integrating vision sensors with other avionic sensors, unmanned systems can autonomously perform a variety of applications, such as vision-based stabilization [37,38], air-to-air tacking [39], navigation in complex environments [40], vision-based pose estimation, and autonomous landing [41,42] as well as localization and mapping [43,44]. Generally speaking, those applications can be roughly divided into two categories, depending on how to use the extracted vision information:

1. Vision-aided target acquisition: Vision sensing is used to search and identify obstacles or targets in the surrounding environment and estimate their relative distance and orientation to the UAV. The estimated information is used to guide the UAV to avoid the obstacles or keep tracking certain targets. In such applications, the motion of the UAV is normally known, and the main challenges are the vision-based object detection and robust tracking algorithms.
2. Vision-aided motion estimation: In contrast to target acquisition, vision sensing is used to estimate the motion of the UAV itself, such as position, velocity, and heading, by fusing vision measurement with the aforementioned navigation sensors and a predefined map. In mapless situations, vision-aided SLAM algorithms need to be utilized to generate the map at the same time. Feature extraction methods in near homogenous environments and data fusion algorithms among different sensors still need to be further investigated.

A single camera is able to handle applications of vision-aided target identification and tracking. When it comes to situations in which relative distance measurement is required, the recommended reliable solution is to utilize stereo vision that is able to provide complete 3-D information. However, stereo vision requires extensive computation and also needs a large baseline between cameras. To solve this problem, we can still use the monocular camera solution for 3-D sensing, which requires relatively lower complexity in terms of hardware and computation than stereo vision as long as the geometry information of the targets or environments is known or partially known.

The performance of vision sensors is greatly affected by environmental conditions, especially lighting conditions. According to operating conditions, vision sensors can be divided into three categories: daylight cameras, night cameras, and thermal cameras. Daylight and night cameras work on the same principle of generating an image based on reflected visible light energy. The major difference is that night cameras greatly magnify small amounts of visible light in dark environments. Unlike daylight or night cameras, thermal cameras produce an image by detecting thermal radiation, and they are able to tell differences in heat as small as 0.01°C. Therefore, they can work in various weather or lighting conditions, such as night, light fog, rain, and smoke.

11.2.1.6 RGB-D Camera

Despite the common usage of monocular or stereo cameras on UAV platforms, the depth camera is a more advanced vision sensor, which can also be used to solve UAV navigation and environmental mapping problems. A depth camera is also called a range camera, flash lidar, or RGB-D camera. Different from traditional cameras, which can only provide 2-D color information, the depth camera also contains per-pixel depth information. There are two main types of depth cameras, namely the time-of-flight (ToF) cameras and the more recently invented projected-light cameras.

Most ToF cameras are very expensive, costing around US$10,000 each. Hence, they may not be suitable for small-scale robotics projects with limited budgets. In contrast, the new generation of consumer depth cameras based on the projected-light technology is much cheaper. Some of them cost only around US$150. Typical examples include the Microsoft Kinect and the Asus Xtion PRO LIVE. The per-pixel depth sensor technology used in these projected-light cameras was first developed by PrimeSense. The technology is patented (United States Patent US7433024). These sensors are able to project a known infrared speckle pattern, which can then be captured by an infrared camera and compared part-by-part to the reference patterns, captured at known depths and stored in the device previously. An algorithm runs in the device to estimate the per-pixel depth based on the reference pattern that the projected pattern matches best. The depth data is then associated with a precalibrated RGB camera and yields a point cloud in 3-D space, with which each point has also color information associated. In addition, approximated surface normals are sometimes calculated and stored with each point.

Commercially available depth cameras have their respective advantages and disadvantages in aspects such as range, resolution, frame rate, field of view, weight, dimension, and so on. Table 11.4 lists the key specifications of four popular depth cameras, which can possibly be used on UAV onboard systems. In spite of their specification differences, they have one common shortcoming, which is the drastic drop in performance when used in outdoor environments. Due to sunlight interfering with the infrared light source, most depth cameras simply fail in outdoor environments. This is expected as these products were originally developed for indoor electronic gaming purposes. Nevertheless, companies such as PMDTec are doing investigations on ambient light suppression technology to ensure more robust performance, which can possibly solve the problem of outdoor usage.

Depth cameras have been utilized in various robotics and UAV-related projects. In [45], the consumer RGB-D cameras and their applications are reviewed. It is mentioned that consumer RGB-D cameras have been used for applications including RGB-D mapping [46,47], interactive 3-D modeling of indoor environments [48], UAV autonomous flight [49,50], and real-time visual and point cloud SLAM [51]. SLAM based on a depth camera has become popular in recent years. In [52], an RGB-D SLAM system is proposed and evaluated via a large set of sequences of different indoor scenes with varying camera speeds and illumination conditions.

To work with consumer RGB-D cameras, several software libraries can be used. They include the Point Cloud Library (PCL), the Microsoft Kinect SDK, and OpenCV. PCL handles data structure, filters, segmentation, registration, visualization of 3-D point clouds,

TABLE 11.4

Comparison between Different Depth Camera Products

	PMD [vision] CamCube 3.0	**Mesa Imaging SR4000**	**Microsoft Kinect**	**Asus Xtion PRO LIVE**
Type	ToF	ToF	Projected-light	Projected-light
Range (m)	7.5	0.1–10	0.8–4	0.8–3.5
Resolution	200 × 200 or 160 × 120	176 × 144	640 × 480	640 × 480 or 320 × 240
Frame rate	40 or 80	50	30	30 or 60
FOV (°)	40 × 40	43.6 × 34.6	57 × 43	58 × 45
Weight (g)	1438	470	440	228
Dimension (cm)	Not published	65 × 65 × 68	24.9 × 6.6 × 6.7	18 × 3.5 × 5

and input–output interface with Microsoft Kinect, Asus Xtion, and other RGB-D sensors based on the PrimeSense technology. The Microsoft Kinect SDK only works for Kinect, and it is focused more toward human body action identification. OpenCV has abundant useful functions to deal with 2-D image processing and some 3-D data interpretation.

11.2.2 Computers

The primary functions of onboard computers include analyzing and processing various data delivered by onboard sensors, executing missions, communicating with the ground station and other UAVs, and logging flight data for post-flight analysis.

To select suitable onboard computers, special attention should be paid to size, weight, input/output (I/O) port configuration, expendability, antivibration property, and power consumption.

A single board computer (SBC) is still the first choice for UAV systems as they have a compact size and the complete features of a fully functional computer, including microprocessor(s), memory, input/output, storage, and so on. Especially, PC-104–based SBCs are strongly recommended because they are designed for harsh environments from the ground up. For instance, a UAV system using a PC-104–based flight control computer has been reported in [53].

In order to implement sophisticated vision algorithms, powerful and advanced SBCs have been widely used [53–55], including the popular PC/104(-plus)–based SBCs and other powerful SBCs, such as the Mastermind computer from Ascending Technologies and fit-PC2i from fit PC. For example, a vision system using a Lippert CoreExpress 1.6 GHz Intel Atom board with a WiFi link has been proposed in [56]. Feature detection and frame-to-frame motion estimation algorithms were implemented to realize autonomous navigation of a quadrotor UAV in indoor environments. On the other hand, micro UAVs expect ultralight and small-size onboard computers, such as computer-on-modules (COMs). The authors in [57] presented a vision system they developed based on a Gumstix Overo Fire with 600 MHz and a webcam to realize vision-aided indoor navigation. A novel and efficient vision algorithm was proposed to realize the robust landmark tracking and path generation onboard. The flight tests verified the robustness and efficiency of the proposed system. In addition to SBCs, it is also useful to implement vision algorithms in special embedded systems such as digital signal processors (DSPs) and field-programmable gate arrays (FPGAs) to speed up the processing, but these technologies require long development time and special development technique compared with the general purpose processors.

11.2.3 Actuator Management

Actuator management is to realize smooth switching between the manual control mode and the automatic control mode. The requirements for the actuator management are listed as follows:

1. Reliable switching function: The switching between automatic control and manual control should be fast and smooth.
2. Sufficient input/output channels: For most aircrafts, five onboard servos are equipped to drive the aircraft. Adding an extra channel for switching function and some necessary redundancy, the channel number must not be less than seven.

3. Capability of recording the servo actuator's input signal: This function is important in initial manual flight tests. The recorded input data are essential for deriving the dynamic model of the UAV and for evaluating control performance.
4. High resolution: Substantially, the input-recording and servo-driving functions are the A/D and D/A procedure. The resolution should be sufficiently high to ensure the data consistency and accuracy.

11.2.4 Communication Unit

The communication units in the UAV system framework are deployed as interfaces between the UAV entity itself and external entities. The external entity can be the GCS for the ground operator or another UAV entity for information exchange. With UAV-to-GCS communications, the operator can remotely control and monitor the UAVs in operation. With inter-UAV communications, the UAV team can multiply their capability and effectiveness in cooperative tasks. Generic communication architecture is illustrated in Figure 11.6.

Although various types of communication devices are available on the market, they can be categorized into two categories based on the device protocols. One is the commonly used UART protocol–based devices. The other one is the TCP/IP protocol–based devices.

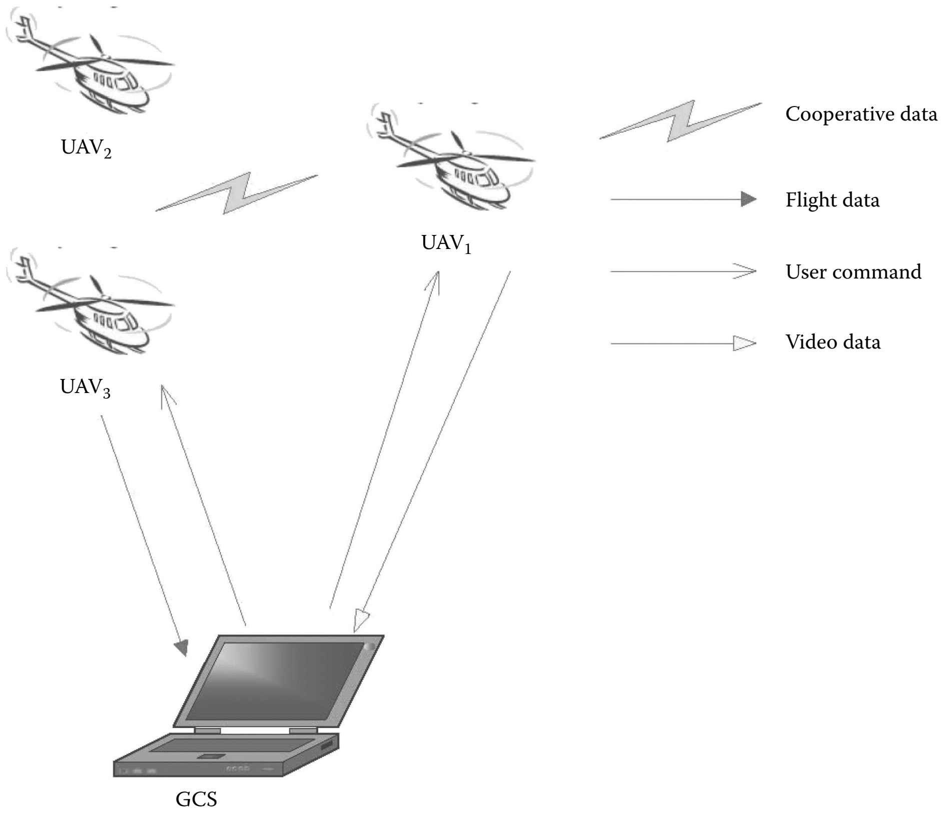


FIGURE 11.6
Communication architecture in a multiple-UAV system.

TABLE 11.5

Communication Device Specifications

Communication Module	Data Bandwidth	Range	Protocol
UART interface	115,200 bps	32 km	UART
802.11g	54 Mbps	250 m	TCP/IP
3G modem	300 kbps	100 km	TCP/IP
4G modem	5 Mbps	100 km	TCP/IP

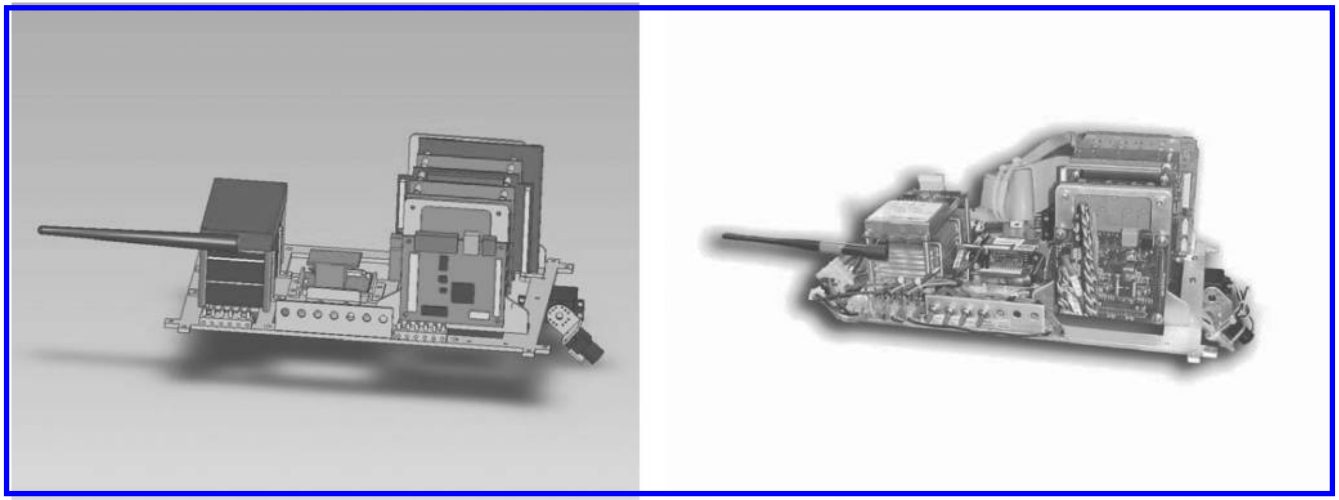
The UART devices are mostly used for long range and low data bandwidth communication. TCP/IP enabled wireless devices are utilized for high data bandwidth communication, such as 802.11 a/b/g/n modules and telecommunication modules with 3G/4G functions. Equipped with such a high bandwidth communication unit, both status data and onboard video can be transmitted in real time. The typical values of data bandwidth and communication range of those communication devices are summarized in Table 11.5. We observe that the 4G devices can deliver both high data bandwidth and long distance coverage at the same time but at the cost of a relatively high subscription fee to local telecommunication service providers. On the other hand, the 802.11-based devices can work in ad hoc mode without any communication infrastructures compared with the 3G/4G devices.

However, once a UAV flies at low altitude or into cluttered environments, such as an urban canyon, the communication range and bandwidth will decrease dramatically due to no line of sight (LOS) between the transmitting antenna and the receiving antenna. A possible solution is to use a second high-fly UAV as a relay, which can extend the range substantially. Of course, the relay UAV introduces additional complexity, such as self-jamming, antenna placement, and the dynamic mission planning for UAVs to serve as communication relays [7].

11.2.5 Hardware Integration

Based on the hardware components selected, we now proceed to carry out systematic integration of those components to achieve reliable performance.

1. Computer-aided virtual design: The lack of a powerful 3-D design environment causes great difficulty in layout design and the integration of hardware components. As a result, the design and integration procedure has to be iterated quite a number of times, which prolongs the total construction time. To avoid such problems, a powerful virtual design environment should be adopted, such as SolidWorks. In the virtual design environment, the virtual counterpart can be modeled to be identical to the real hardware component, both in shape and material properties. When the 3-D design is finished, the corresponding 2-D views will be generated at the same time for the convenience of mechanical manufacturing. Layout design for the avionic system can be realized in the virtual design environment to achieve near optimal CG balance and space usage. One example of virtually designed avionics and its real counterpart is illustrated in Figure 11.7. In addition to the layout design, the simulation environment is also used for the mechanical structure design and modeling. For example, landing gear for a UAV helicopter has been analyzed in SolidWorks to simulate its strength and stiffness (shown in Figure 11.8) to reduce the need for costly prototypes or design changes later on.



(a) (b)

FIGURE 11.7
An example of (a) virtually designed avionics and (b) real avionics.

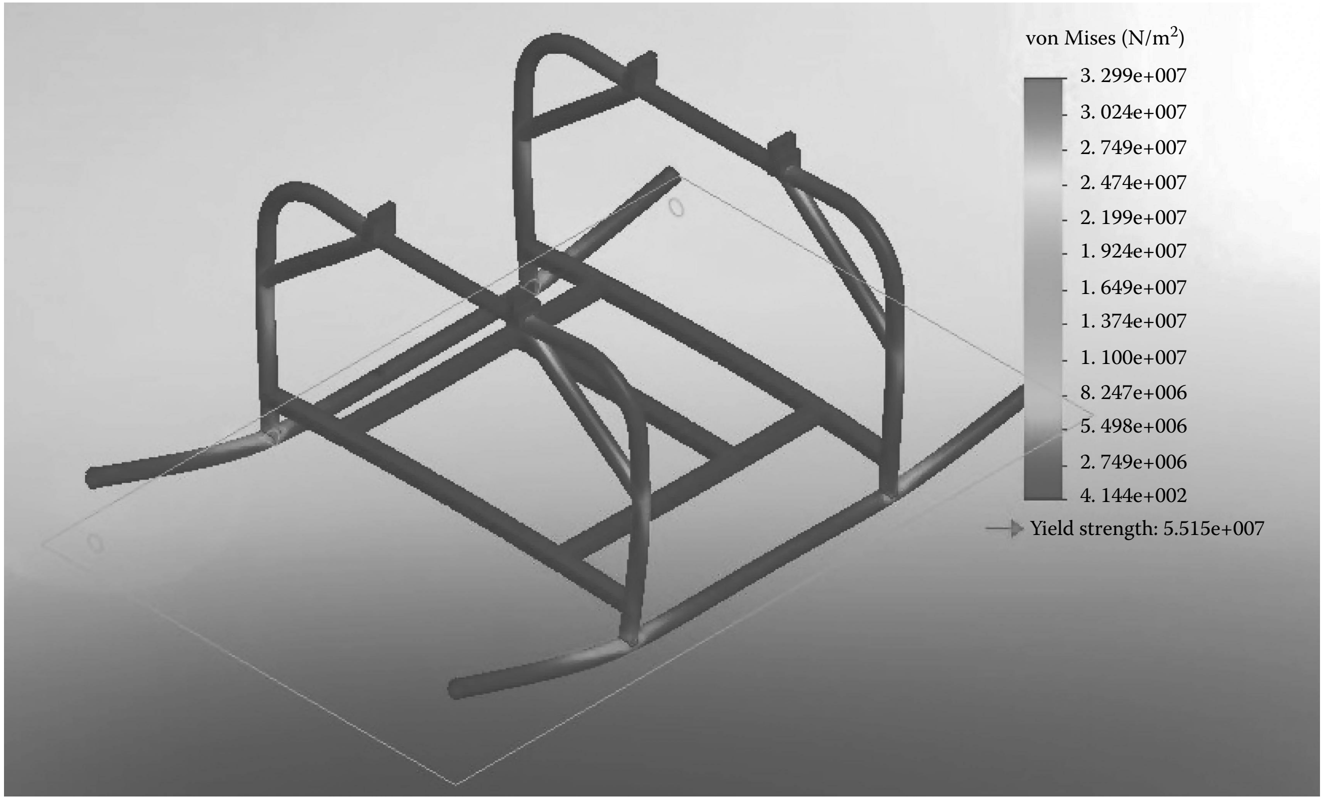
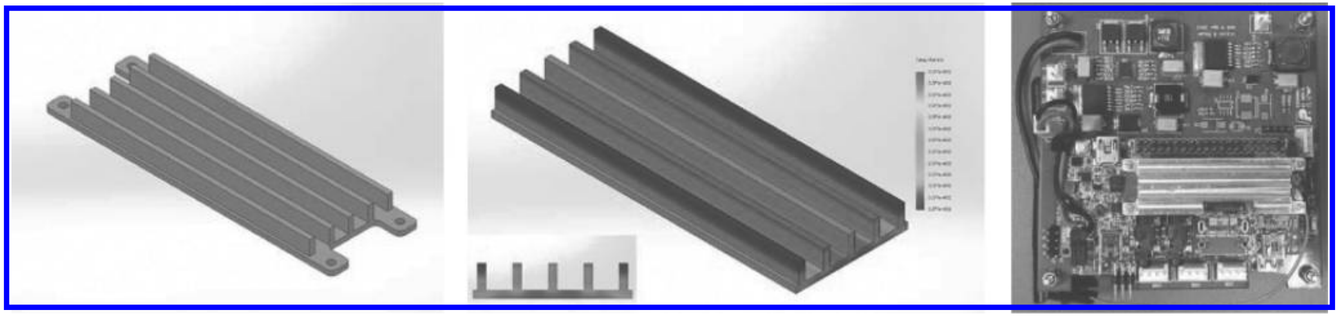

von Mises (N/m^2)
3. 299e+007
3. 024e+007
2. 749e+007
2. 474e+007
2. 199e+007
1. 924e+007
1. 649e+007
1. 374e+007
1. 100e+007
8. 247e+006
5. 498e+006
2. 749e+006
4. 144e+002
Yield strength: 5.515e+007

FIGURE 11.8
Structure analysis of landing gear.



(a) (b) (c)

FIGURE 11.9
Heat sink designed for an ARM-based processor: (a) virtual design, (b) temperature distribution, and (c) installed.

2. Antivibration design: For any small-scale UAVs, there are several main vibration sources that should be taken into account: the rotation of the rotors or propellers and the power engine. It has the potential to introduce bias to measurement data and to cause loss of connection of mechanical components. For this reason, an antivibration design is necessary to ensure the overall onboard system is working properly and reliably.
3. Power supply design: The main consideration in the power supply design is to meet the overall experimental requirements and system safety. For example, although a single battery is sufficient to power the avionic devices, two batteries have been chosen instead to enhance the overall safety of the system. Thus, the system can still run smoothly and guarantee manual maneuvering even if one of the batteries is out of power.
4. EMI shielding design: Electromagnetic interference (EMI) is a serious issue for small-scaled UAVs as all of the highly integrated electronic components are required to be mounted in a very limited space. The main problems arising from EMI include reducing the effective range of RC manual control, generating errors in INS/GPS measurements, and causing data losses in wireless communications. These problems have to be eliminated or reduced to a minimum before conducting actual flight tests. It is suggested to use aluminum boxes and foil to isolate the necessary electronic components.
5. Thermal analysis: All the electronic components have a temperature-dependent property. It is a necessary consideration to design a guard over components that can easily overheat, such as processors and communication units. For instance, we have designed a heat sink to protect an ARM-based computer system operating in outdoor high temperature environments, shown in Figure 11.9. This design can effectively reduce the temperature of the processor and ensure the continuous operation of the computer system in high-temperature environments.

11.3 Unmanned System Software

A generic framework of unmanned system software is presented in Figure 11.10. In this framework, the functions of all necessary UAV modules, including the onboard system and ground control system are clearly presented. The logical data flows among different modules facilitate the design of UAV systems.

11.3.1 Onboard Real-Time Software System

The main functions of the onboard real-time software system are to (a) collect sensor data, (b) process and fuse them, (c) feed the control law, and (d) send the servo driving signals to realize desired automatic and intelligent operations. Meanwhile, the UAV status data is transmitted back to the GCS and exchanged among team members. The UAV onboard software system has the following functions.

1. Sensing and measurement: The sensing data may come from different sources, such as the IMU, GPS, ultrasonic, laser scanner, or vision system, depending on the configuration of the avionics. All the sensing data are fed into the data selection

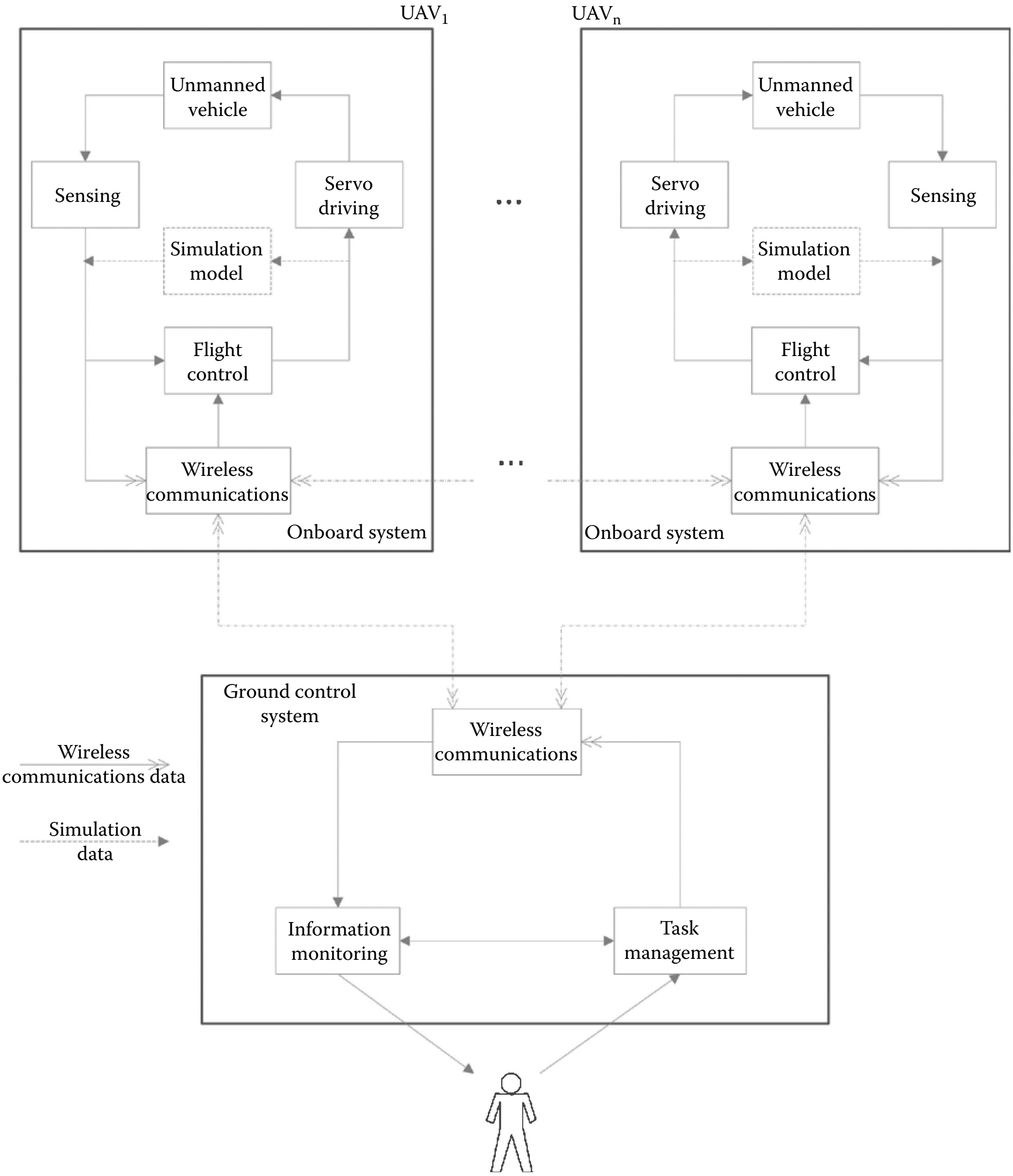

UAV$_1$
UAV$_n$
Unmanned vehicle
Sensing
Servo driving
Simulation model
Flight control
Wireless communications
Onboard system
…
Unmanned vehicle
Servo driving
Sensing
Simulation model
Flight control
Wireless communications
Onboard system
Ground control system
Wireless communications
Information monitoring
Task management
Wireless communications data
Simulation data

FIGURE 11.10
Framework of UAV systems.

and the processing unit, which will be selected and combined for the flight control and other mission algorithms.

2. Flight control system: The flight control module consists of three main units: task scheduling, outer-loop control, and inner-loop control. The task scheduling is to generate the outer-loop references given current status data and user commands from wireless communications. Based on the reference signals, the outer-loop realizes position, velocity, and heading control by generating references for the inner loop. The role of the inner loop is to stabilize the attitude of the UAV.
3. Servo driving: During the control procedure, the outputs are sent to the servo-driving module to drive the actuators on the UAV.
4. Communication: The communication block is to transfer the status data and commands between the vehicle and the GCS or between vehicles. The status data incorporates the flight information, such as position, velocity, and attitude. The ground operator can issue predefined commands, such as takeoff, hover, cruise flight, and landing to UAVs to perform specific tasks. The onboard video stream is also needed to feed back to the GCS. In addition, the inter-vehicle communication is applied for cooperative data exchange to achieve UAV team cooperative control, such as formation flight control.
5. Data logging: The key data, such as sensor measurements and control signals, in the flight will be logged for further analysis. The data logging is usually designed as a background task to minimize the interference to other critical tasks.
6. Emergency function: The emergency function is learned from a couple of crash incidents of our UAV helicopters. There are many causes for UAV failures, such as drastic changes in environment or hardware and software failure. To handle such emergency situations, the control task thread checks all sending data at every cycle before applying control actions. Once an abnormality is detected, the emergency control function will be activated immediately to send an alert signal to the GCS to inform the pilot to take over the control authority, drive and maintain all the control outputs to their trimmed values in the hovering condition, and slow down the engine or motor speed if the control authority is still at the automatic side after a predefined alert time. The flight data will also be recorded, which is important for fault analysis.
7. Simulation model: The simulation model block is an offline module used to conduct hardware-in-the-loop simulation before flight tests, which can generate the dynamics of the UAV based on a mathematic model. The UAV model can be formulated as an order ordinary differential equation (ODE) as below:

   $$\dot{x} = f(t, u, v, x), \tag{11.1}$$

   where x represents the UAV model output with multiple states, t is the current system time, u is the current control signal input with multiple channels, and v is the wind disturbance in three directions. For the ODE implementation, the classical Runge-Kutta approximation method is applied in the software.
8. Main function: To manage all above tasks and functions with predefined priorities and properly assigned time slots.

Although it is possible to implement a software system using multiple nested loops to handle the above tasks in real time without an operating system, such a software system is not scalable and reliable for complex missions. The development of avionic software systems should preferably to be carried out in a real-time operating system (RTOS) environment, which can guarantee the final system is executed in deterministic behavior, based on certain scheduling, intertask communications, resource sharing, interrupt handling, and memory allocation algorithms [58]. Currently, the three most popular real-time operating systems adopted in the UAV development are the QNX Neutrino [59], VxWorks [60], and RTLinux [61]. In our current software system, the QNX is adopted for the mission-critical control purposes to meet strict timing deadlines, and the Linux is adopted for computation intensive mission planning tasks.

11.3.2 Ground Control Software System

The main responsibility of the ground control software system is to establish effective communications between the avionic system and the ground users and pilots. To fulfill this aim, the ground station is generally required to have the following fundamental capabilities: (a) displaying and monitoring the inflight status; (b) displaying images captured by the onboard system; (c) generating and updating flight trajectories; (d) sending control commands to the avionic system; (e) facilitating the ground piloted control or automatic control, especially in sudden occurrences of urgent events, such as emergency landing and cruise; and (f) logging inflight data. Other features, such as displaying the reconstruction of the actual flight status in a 3-D virtual environment, can be very helpful to the ground users when the UAV is flying out of sight (see, for example, [62]).

Compared with its avionic counterpart, the real-time feature for the ground station software system is preferable but not strictly compulsory. As such, many ground station software systems, particularly for scientific research and commercial purposes, are not developed under a RTOS environment. Instead, other powerful programming environments with rich interface capacities, such as Windows-based Visual C++ [63], are commonly adopted.

11.4 Case I: Design of a Coaxial Rotorcraft System

In the following section, we will detail the mechatronics design of unmanned aircraft systems using a couple of examples. The first example is a fully functional unmanned rotorcraft system: GremLion [64]. GremLion is a new small-scale UAV concept using two contra-rotating rotors and one cyclic swashplate. It can fit within a rucksack and be carried easily by a single person. GremLion is developed with all necessary avionics and a ground control station. It was employed to participate in the 2012 UAVForge competition.

11.4.1 Hardware System

GremLion, shown in Figure 11.11, features a coaxial design driven by two contra-rotating rotors that can compensate the torque due to aerodynamic drag. Such a design allows for a more stable, more maneuverable, quieter, and safer helicopter due to the inclusion of a coaxial main rotor and exclusion of a tail rotor, which results in a smaller footprint.

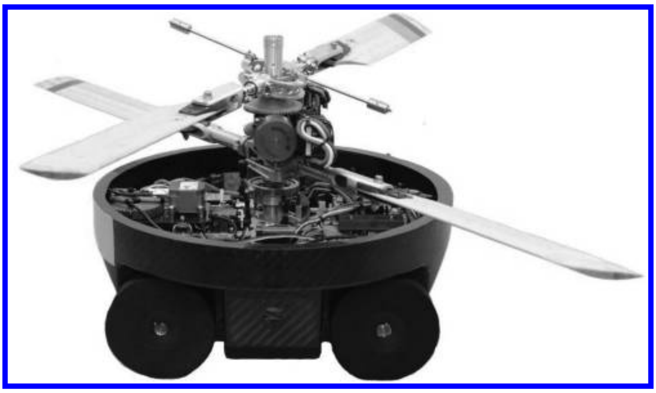


FIGURE 11.11
The GremLion UAV.

Coaxial helicopters also provide a better thrust-to-weight ratio than traditional helicopters, produce greater lift, and are also much more efficient [65]. Therefore, this platform is suited for the size requirement of the competition, which is required to be kept in a rucksack. The key specifications of the platform are listed in Table 11.6.

To reduce the complexity of the actuation system of the conventional coaxial design, a novel actuation system has been employed in GremLion, which is shown in Figure 11.12. The operating principle of this actuation system is presented as follows:

1. Unlike conventional single-rotor helicopters that utilize the collective pitch of their rotor blades to adjust the lift force, GremLion's rotor pitches are fixed, and the thrust variation is accomplished by changing the rotor spinning speed simultaneously. Hence, the vertical motion is controlled by the pulse width modulation (PWM) signals fed to the motors attached to the top and bottom rotors. As illustrated in Figure 11.12, the throttle input δ_{col} can adjust the speed of the upper rotor and the lower rotor simultaneously.
2. The helicopter yaw motion (head turning) is produced by the difference of spinning speed between the top and bottom rotors. When one rotor spins, other than the lifting force it creates, it also generates a rotational torque on the fuselage of the helicopter in the direction opposite to the rotor spinning direction. Note that the top and bottom rotors always spin in opposite directions so that their torques

TABLE 11.6
Specifications of the GremLion UAV

Specifications	GremLion
Upper rotor span	798 mm
Lower rotor span	895 mm
Upper rotor speed	1900 rpm
Lower rotor speed	1700 rpm
No-load weight	2.4 kg
Maximum takeoff weight	5.1 kg
Power source	LiPo battery
Flight endurance	15 min

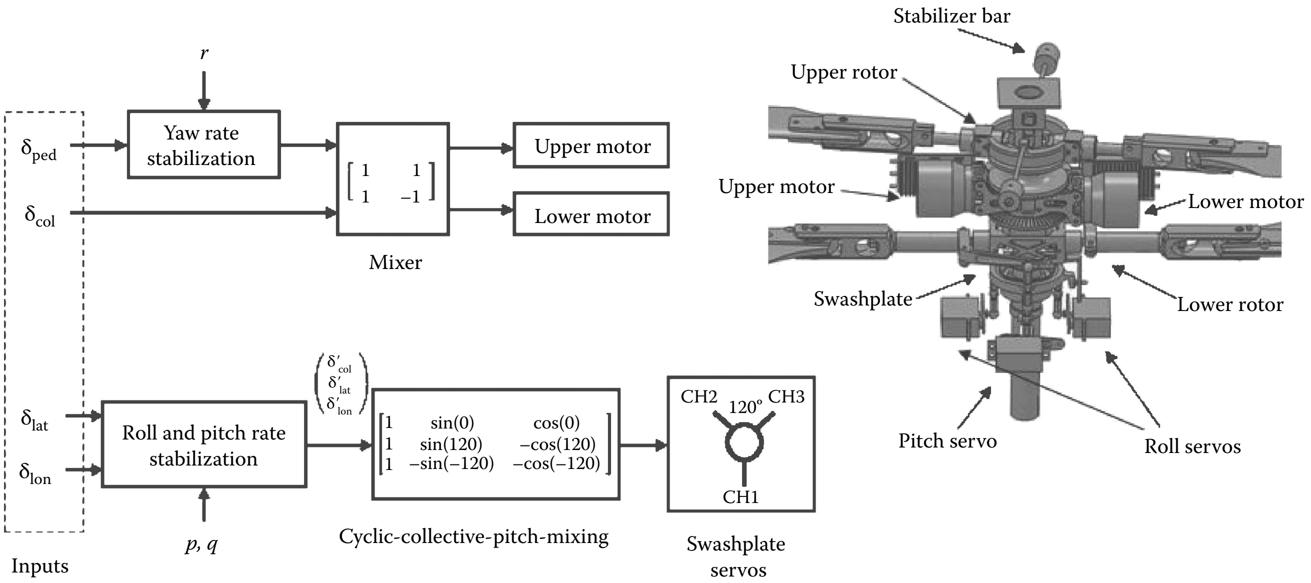

r
Yaw rate stabilization
δ_{ped}
δ_{col}
$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Mixer
Upper motor
Lower motor
δ_{lat}
δ_{lon}
Inputs
Roll and pitch rate stabilization
p, q
$\begin{pmatrix} \delta'_{col} \\ \delta'_{lat} \\ \delta'_{lon} \end{pmatrix}$
$\begin{bmatrix} 1 & \sin(0) & \cos(0) \\ 1 & \sin(120) & -\cos(120) \\ 1 & -\sin(-120) & -\cos(-120) \end{bmatrix}$
Cyclic-collective-pitch-mixing
CH2
120°
CH3
CH1
Swashplate servos
Stabilizer bar
Upper rotor
Upper motor
Lower motor
Swashplate
Lower rotor
Pitch servo
Roll servos

FIGURE 11.12
Operating principle of GremLion.

cancel each other. In order to make the heading of the helicopter stable, a hardware rate gyro is installed to finely adjust the spinning speed of the two rotors so that yaw dynamics become much more damped. As shown in Figure 11.12, the rudder input δ_{ped} for control of the yaw of the vehicle differentiates the spinning speeds of the two rotors.

3. In order to have lateral and longitudinal motions, the bottom rotor cyclic pitch is actively controlled by three servos. This is done through a swashplate mechanism, which acts as a link between the servos and the bottom rotor cyclic pitch. As shown in Figure 11.12, the aileron input, δ_{lat}, controls the leftward and rightward tilting motion of the swashplate. Such a movement changes the cyclic pitch angle of the lower rotor blades and results in both a rolling motion and lateral translation. The elevator input, δ_{lon}, is responsible for the forward and backward tilting motion of the swashplate. This tilting also changes the cyclic pitch angle of the lower rotor blades but results in pitching motion and longitudinal translation. The aileron and elevator inputs cooperate with the roll and pitch rate feedback controller to stabilize the angular rate of roll and pitch motions. Such a rate feedback controller is used to allow a human pilot to control the oversensitive dynamics of the bare platform.
4. The upper rotor is equipped with a stabilizer bar to further increase the stability of the vehicle. The top rotor is not actively linked to any servos, but it is passively controlled via a mechanical stabilizer bar. With the presence of this stabilizer bar, the top rotor always has a cyclic pitch (with respect to the body frame) countering the inclination of the fuselage at any single moment. This slows down the whole platform's response to the rapid changes in the cyclic pitch of the bottom rotor [66]. In this way, the helicopter stability is increased, but the maneuverability is decreased.

11.4.1.1 Navigation Sensors

IG-500N (see Figure 11.13) is one of the world's smallest GPS-enhanced AHRS embedded with an extended Kalman filter (EKF). It includes a MEMS-based IMU, a GPS receiver, and a pressure sensor. It is able to provide precise and drift-free 3-D orientation and position even during aggressive maneuvers, updated at 100 Hz. Its key specifications are summarized in Table 11.7.

11.4.1.2 Computers

The onboard processor is the brain of the whole avionic system. It collects measurement data from various sensors, performs sensor filtering and fusion, executes flight control law, and outputs control signals to carry out the desired control actions. In addition, it is also responsible for communicating with the GCS for real-time inspection and command issuing as well as logging in-flight data for post-flight analysis. Hence, selecting suitable COTS processors is crucial to ensure successful implementation of the UAV system. We have chosen two Gumstix Overo Fire embedded computers for flight control and navigation purposes, respectively (see Figure 11.13). This embedded computer system has a main processor running at 720 MHz and a DSP coprocessor. The main processor is an OMAP3530 ARM processor from Texas Instruments, and it is one of the fastest low-power embedded processors as of this writing. Moreover, it has WiFi functionality despite its tiny size

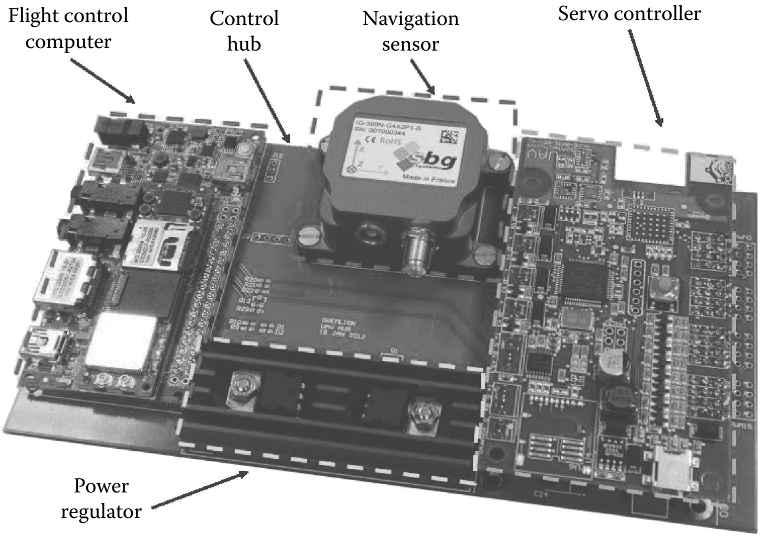


FIGURE 11.13
Control hub with all the modules.

TABLE 11.7
Main Specifications of IG-500N

Specifications	IG-500N
Attitude range	360° in three axes
Attitude accuracy	±0.5 (pitch, roll), ±1 (heading)
Accelerometer range	±5 g
Gyroscope range	±300
Magnetometer range	±1.2 Gauss
GPS accuracy in CEP	2.5 m (horizontal), 5 m (vertical)
Output rate (Hz)	{1, 25, 50, 75, 100} selectable
Dimensions	36 × 49 × 22 mm
Weight	46 g (with aluminum enclosure)
Power consumption	550 mW at 5.0 V

and light weight. In order to improve its real-time performance, the original Linux operating system provided by the manufacturer is replaced by the QNX Neutrino. Custom-built autopilot software developed by the NUS UAV Research Group is used to realize the desired autonomous flight control.

11.4.1.3 Servo Controller

An eight-channel PWM servo controller, UAV100 (see Figure 11.13), is adopted to allow servo outputs to be controlled by an onboard computer or control command from the RC receiver, depending on the state of a switching signal from the RC transmitter. Although GremLion maneuvers autonomously in the air, it is desirable to have a failsafe feature to allow the ground pilot to take over control during emergencies. This servo controller allows the pilot control to take over the control of a UAV at the flick of a switch of the transmitter to prevent a catastrophic incident from a malfunction in the flight computer. This function gives us the ability to test flight control software early without fear of damaging a test vehicle.

11.4.1.4 Communication

The communication unit includes a pair of Microhard wireless data transceivers. This pair of transceivers establishes communication between the onboard system and the ground station. They are configured to operate in a point-to-point mode and work in 2.400 to 2.4835 GHz. The transceiver used in the onboard system is set as a point-to-point slave and connected to the flight control computer board. The transceiver in the ground station is set as a point-to-point master and connected to a laptop.

In addition to the RF transmission, another data transmission mechanism is based on the 3G network that possesses continental distance (around 10 s of km) and high data bandwidth (100~300 kpbs). To facilitate the image transmission, the images are compressed via a JPEG compression method provided by OpenCV libraries. The size of the compressed image size is about 3K bytes that can be reasonably accommodated by the 3G network bandwidth. The update frequency is achieved as 2 Hz. The 3G module adopted is the UC864-E from Telit Company, which is well compatible with the Gumstix board. The communication protocol for image transmission is selected as TCP that can guarantee reliable image transmission although some delays can be expected for the handshaking and image retransmission.

11.4.1.5 Control Hub

The control hub, shown in Figure 11.13, is a motherboard designed to host subsystems for control purposes. It has the following features:

1. Module connection. Aforementioned modules, such as the Gumstix board, the IG-500N, and the UAV100 servo control board, are installed on the slots on the control hub and connected to the onboard power regulator and other essential components through the control hub. In addition to the mounting slots, extra mounting holes on the control hub have been used to lock the installed modules to resist the vibration and shock in flight and landing. Manual wire wrap has been minimized to improve the reliability and quality of the system.
2. Level shifter. An onboard level shifter, MAX3232, has been built in the control hub to convert the serial signal from RS-232 level to TTL level, which has been used to make the output of IG-500N compatible with the Gumstix board.
3. Power regulation. To power up all the avionics, linear regulators are built in the control hub to convert a power input from a three-cell LiPo battery into a 5 V output with 10 A capacity and a 2–8 V adjustable output with 10 A capacity. The 5 V output powers the Gumstix board, the rate gyro, and the electronic mixer. The adjustable output powers the servos.

11.4.1.6 Hardware Integration

The final integrated platform is shown in Figure 11.11. In addition to the essential mechanical parts, all the related avionic components have been assembled onto the system. This platform has been extensively used in test flights for model identification and verification. Layout design for onboard computer systems is a challenging issue for small-scale UAVs. In what follows, we propose a simple and unified layout design approach, which is independent of the hardware components used and can be easily adopted to construct any small-scale UAVs.

1. Location of the navigation sensor. The essential rule of this step is to mount the navigation sensor IG-500N with the control hub as close as possible to the center of gravity (CG) of the UAV to minimize the so-called lever effect, which can cause bias on the measured accelerations when the UAV performs rotatory motions. Based on the experiences we gained from the construction of our earlier version UAVs [53], we find that it is easier to control the UAV when the onboard system is mounted underneath the bare vehicle. For such a layout, the general guideline is to line up the CGs of the INS/GPS, the onboard computer system and the basic helicopter along the z-axis of the body frame. Because the CG location of the bare vehicle is fully known using the pendulum test introduced in [67], the mounting location of the navigation sensor in the x-y plane of the body frame can be determined. The offset between the CG of the UAV and that of the navigation sensor is only in the z-axis and is unavoidable. However, it can be minimized by carefully considering the height of the onboard system and adding necessary space between the bare helicopter and the onboard system for bumping avoidance. In addition, the GPS antenna of IG-500N has been located in the top of the main shaft of GremLion to have a good view of the sky in order to obtain a stable signal lock.
2. CG balancing. The locations of the following four components, that is, the vision hub, the onboard camera, the wireless modem, and the battery packs, have to be carefully selected. In general, the camera and the wireless modem are to be mounted at the front part for the convenience of observation and wireless communication. The vision hub is placed on the back to balance the overall CG of the onboard system. The battery packs are placed beneath the fuselage and along the z-axis of the body frame. Furthermore, we also guarantee that the CG of the onboard system coincides with the CG of the INS/GPS, and the onboard system is symmetrical in both longitudinal and lateral directions.

Antivibration for the platform is a key issue that affects performance of the system significantly. The main vibration sources in GremLion come from the two main rotors with the frequency of 33.3 Hz. This frequency is calculated based on the designed main rotor speed at 2000 rpm, which was also verified using a handhold tachometer.

Several solutions have been employed for antivibration purposes: (a) use four wire-rope isolators mounted symmetrically around the CG of the avionic system; (b) employ a redesigned landing skid that has better connections to the platform; (c) replace wooden blades with carbon blades, which have the same airfoil profile and size but with a smooth surface; and (d) configure the cutoff frequencies of the built-in low-pass filters of the IMU at 10 Hz. These proposed antivibration solutions have been demonstrated in flight. The comparison of the vibrations before and after employing the antivibration is shown in Figure 11.14.

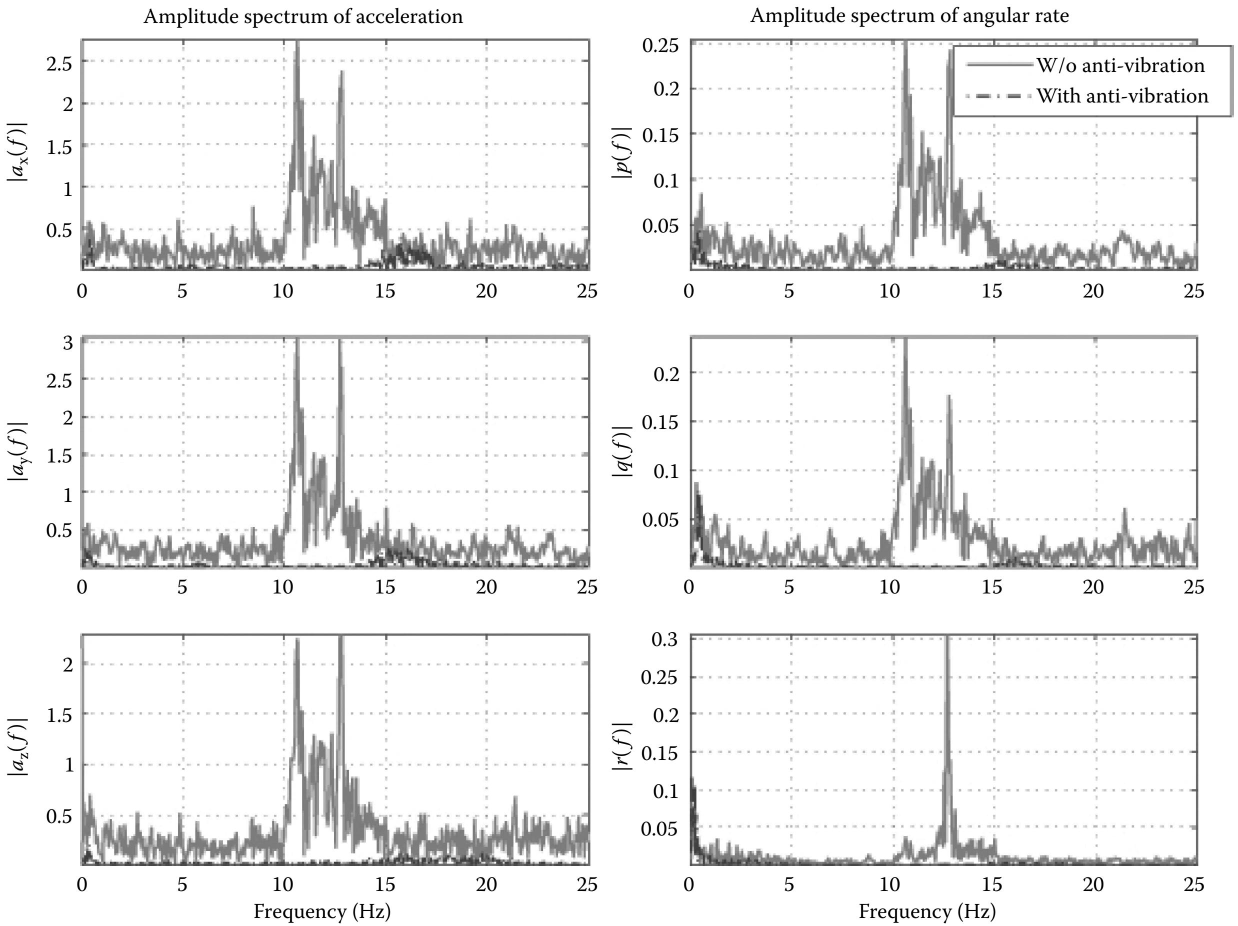


FIGURE 11.14
Comparison of frequency response of the acceleration and angular rate with antivibration solutions.

11.4.2 Software System

11.4.2.1 Onboard Real-Time Software System

The control law implementation is realized via two hierarchical blocks, the inner loop and outer loop as shown in Figure 11.15. The outer loop is to generate references for the inner loop as the input. The flight-scheduling module is to divide a whole flight mission into several specified tasks such as takeoff, path tracking, and landing. The flight mission can be as simple as conducting an automatic hover operation or can be as complicated as surveillance of a group of UAVs. With this hierarchical approach, various high-level missions can be transferred into logical representation and practical implementations. Specifically, for different UAV platforms, the corresponding blocks of the outer loop and the inner loop are activated. In addition, given different control behaviors, such as landing and takeoff, specific control blocks are also developed.

In some critical applications, the reference paths for a UAV must be generated online. This is commonly needed in dynamic environments, such as in the cases of a lost link or waypoint updates from the ground pilot. There are basically two parts: one is the path creation given a certain task, and the other is the outer-loop reference generation from the generated path. In the path creation, the task needs to provide the destination waypoint, which when given the current position and heading, can automatically generated the new path with a specified tracking velocity. The destination waypoint can be uploaded from the Google map with the GPS information or by a user-defined relative distance to the launch position.

The servo-driving block is to control the deflections of actuators according to the control outputs. The servo-driving block has two inputs. One is the manual control signal from the ground pilot, and another input is the automatic control signal generated by the flight control block. One of the two inputs can be selected to output in terms of the switch signal from the manual input. If the manual control is enabled, then the manual input signal will be translated and stored in the data store. Otherwise, the automatic signal will be translated, stored, and finally sent to the servos to realize desired deflection positions. The data logging block is to record the key avionics data in flights, including UAV status, manual and automatic servo signals, user commands, and so on. The recorded data can reflect the working status and property of the whole system and are useful for off-line analysis. Considering the CPU load increase when writing data to the onboard CF card, the data logging thread is executed every 50 cycles, that is, once per second. The above discussed subsystem is represented as an individual task in the whole onboard software system. Based on the theory of systems and control, the above tasks should be executed in

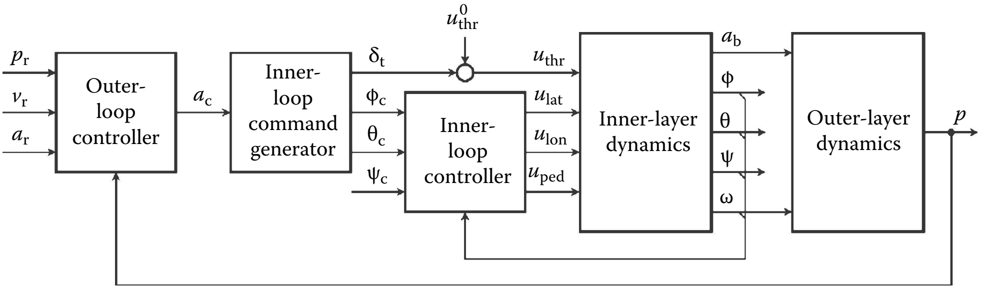


FIGURE 11.15
Dual-loop structure of flight control system.

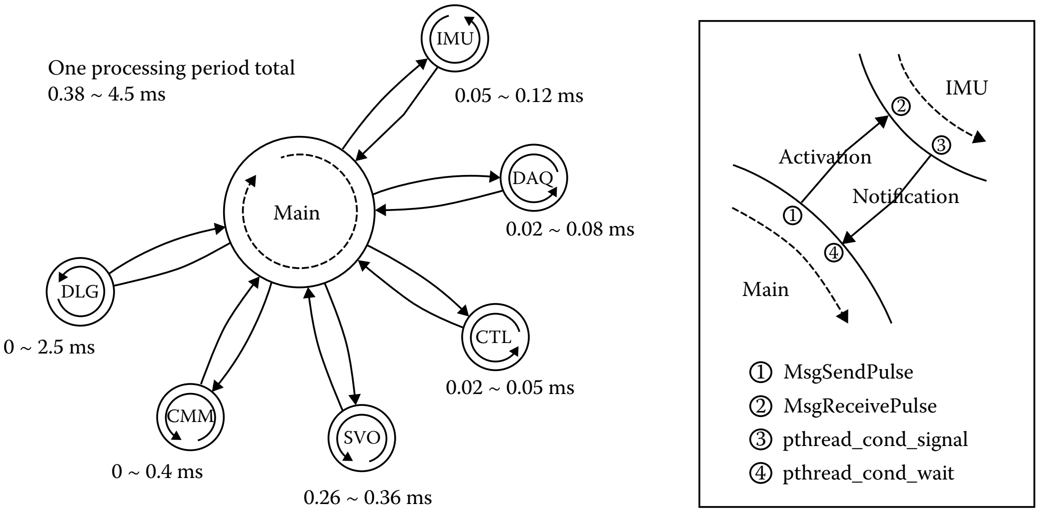


FIGURE 11.16
Task management illustration diagram.

a reasonable order to fulfill the automatic flight control purpose. Considering the context-switching cost, the onboard avionics application is designed in a multithread fashion. Therefore, the above identified task can be accomplished within each working thread. The QNX provides the kernel level mechanism to support the message passing and synchronization of multithread software architecture.

To realize the predefined task execution and synchronization, a task management module is carefully designed, which is shown in Figure 11.16. Note that the tasks involved here are the active tasks that should be executed within every control loop. To achieve fair allocation of processor running time, the round robin scheduling policy is adopted in the multithread design. On the other hand, the background tasks, such as communication receiving (UART-based and TCP/IP-based) are activated once data arrive and if there are still CPU running slices left and the activation mechanism is determined by the scheduling policy in QNX RTOS.

As shown in Figure 11.16, the main program is responsible for activating the next task once the notification message is received from the current running task.

The time deadline is the most important property in real-time systems. As the main control period of the onboard system is set to 20 ms, the timing intervals between each running loop should be examined to test the robustness and efficiency of the avionics system framework. The total processing time for each loop is summarized in Figure 11.17. It can be observed that the timing interval between each loop is separated by strictly following the 20 ms configuration, which provides the fundamental support for correct and stable control law implementation.

11.4.2.2 Ground Control Software System

The GCS is composed of background tasks and foreground tasks. The background layer has mainly two tasks, receiving flight status from and sending commands to UAVs, both of which interact with the UAV onboard CMM task module. The receiving thread

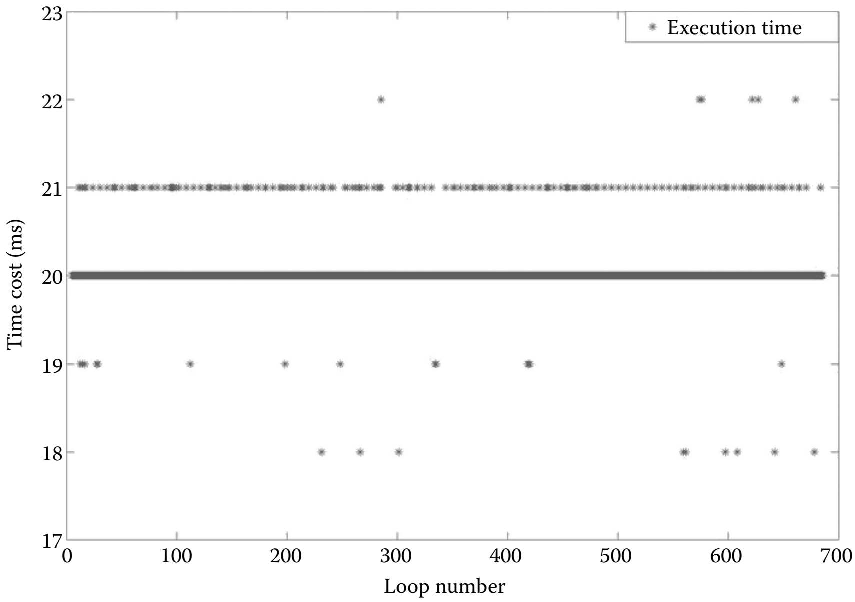


FIGURE 11.17
Time intervals between each loop.

accepts all the data from the fleet of UAVs, and identifies each status data via the telegraph packet header. Consequently, the corresponding multiple display is executed, and the cooperative waypoints of the paths are demonstrated. Similarly, the upload link can broadcast the commands to all UAVs or alternatively send commands to a specific UAV, both via the sending task. The global status data from UAVs are dynamically updated from the background layer. The foreground task is composed of information monitoring and task management, and the information monitoring module consists of various user-friendly views.

A document class implementation in MFC [68] is deployed to realize the communication between the background tasks and foreground tasks. The document class performs the flight data store (up to 2000 updates), data processing (rotation computation in 3-D view), command interpreting and packaging, etc.

Five kinds of views are developed on GCS, including the map view, the curve view, the text view, the command view, and 3-D view, which are all shown in Figure 11.18. To facilitate navigation and better demonstrate the flight trajectories of UAVs, we downloaded from the Internet the map from the Google map server and used the map tiles offline on the flight field conveniently without bothering the Internet service [69]. In the flight test, the GPS data from the onboard system will be updated on the global shared data, and the flight trajectories are drawn correspondingly on the Google map view.

11.4.3 Experimental Results

To verify the developed UAV systems, automatic flight tests have been conducted. Figure 11.19 shows clearly the reference generation onboard with a total of four trapezoids, which represent the four waypoints uploaded by the GCS operator on the Google map. Between every two waypoints, the longitudinal and lateral velocity reference signals increase and

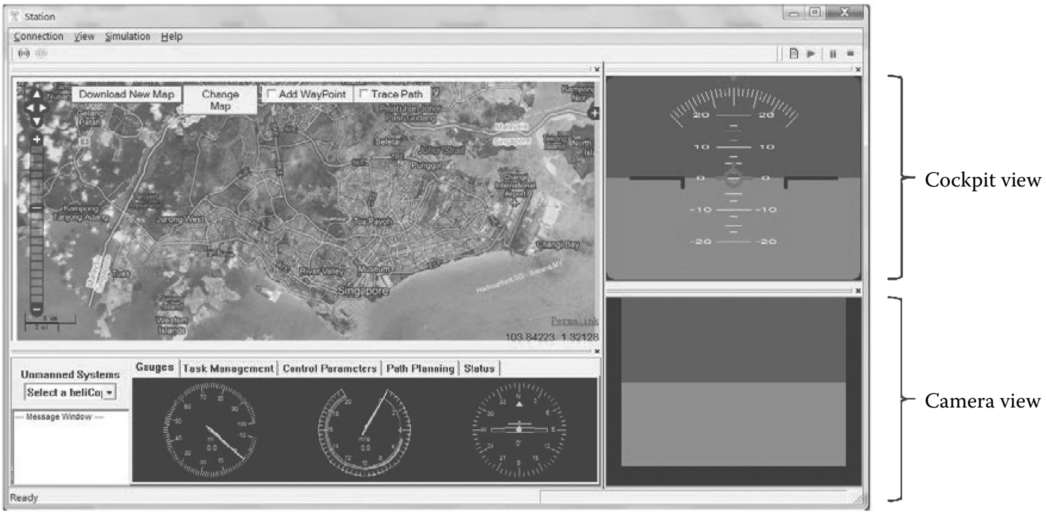

Station
Connection
View
Simulation
Help
Download New Map
Change Map
Add WayPoint
Trace Path
Singapore
Jurong West
Toa Payoh
River Valley
Museum
Changi Bay
103.84223 1.32128
Unmanned Systems
Select a heliCo
Message Window
Gauges
Task Management
Control Parameters
Path Planning
Status
Ready
Cockpit view
Camera view

FIGURE 11.18
Snapshot of GCS layout.

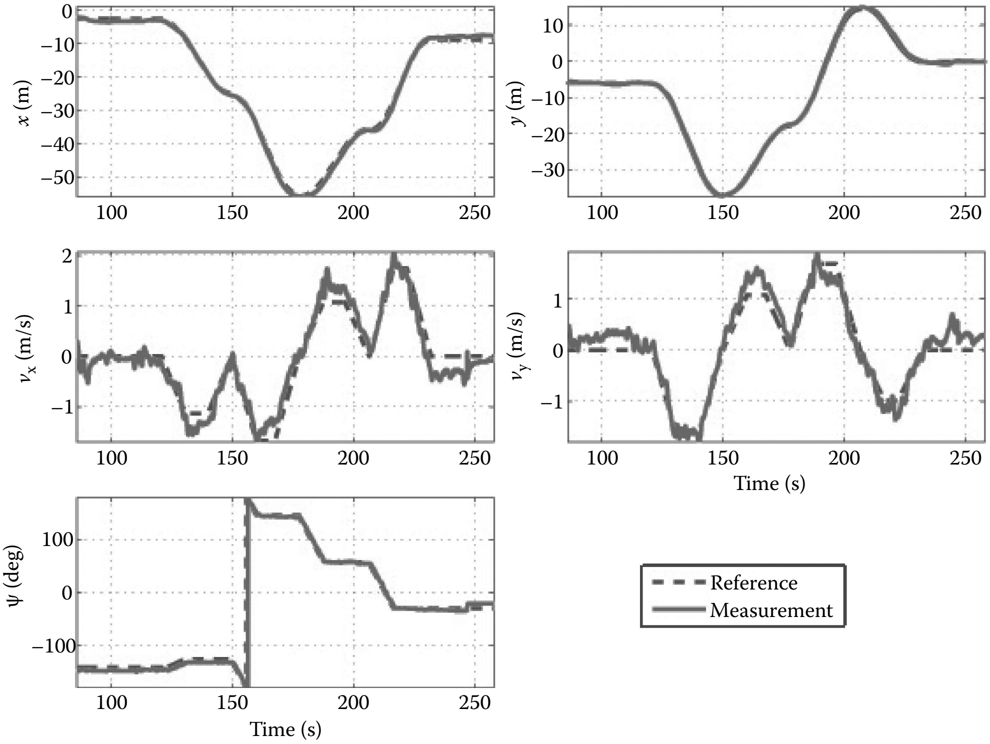


FIGURE 11.19
Experimental results of the automatic flight.

decrease smoothly and the corresponding position reference is also generated correctly. In the flight test, the UAV can automatically track the reference very well.

11.5 Case II: Design of a UAV Cargo Transportation System

In this part, we present a comprehensive UAV cargo transportation system, which involves an innovative cargo grabbing system, a set of UAV autonomous navigation and control algorithms, and a vision-based intelligent cargo searching and guidance method. The proposed UAV system, codenamed NUS^{2}T-Lion (see Figure 11.20) was implemented and had taken part in the second AVIC Cup–International UAV Innovation Grand Prix (UAVGP), which was held in Beijing in September 2013. In this competition, NUS^{2}T-Lion demonstrated its ability of searching and locating cargo on one moving platform and automatically grabbing and transporting the cargo to another moving platform. The whole process from taking off to landing was done without any decision making from the ground operator.

11.5.1 Hardware System

The hardware configuration of NUS^{2}T-Lion follows the universal rotorcraft UAV structure proposed in [1]. As illustrated in Figure 11.21, in which each block represents an individual hardware device, the whole system is constituted by four main parts, namely a bare rotorcraft platform, an onboard avionic system, a manual control system, and a ground control system.

Although the manual control system and the GCS are rather standard for all kinds of UAV systems, the choice of the bare rotorcraft platform and its onboard avionic system is usually application-dependent. For this case, they should be properly chosen and integrated for the UAV cargo transportation task. It is believed that by designing the hardware configuration creatively, difficulties for the later software algorithm development can be minimized.

To realize fully autonomous flight, an onboard avionic system with sensors, processors, and other electronic boards has to be designed. All components used on the NUS^{2}T-Lion

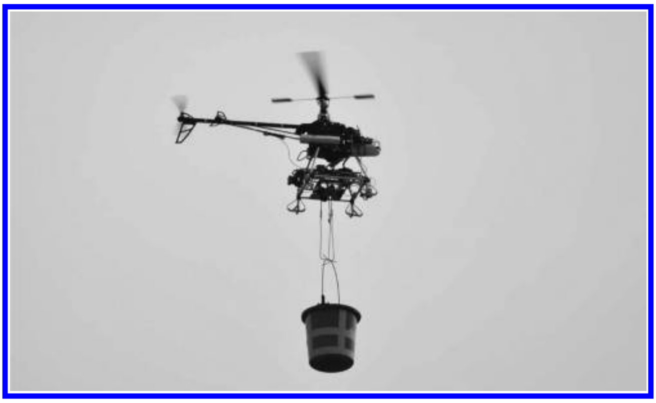


FIGURE 11.20
NUS^{2}T-Lion with cargo grabbing capability.

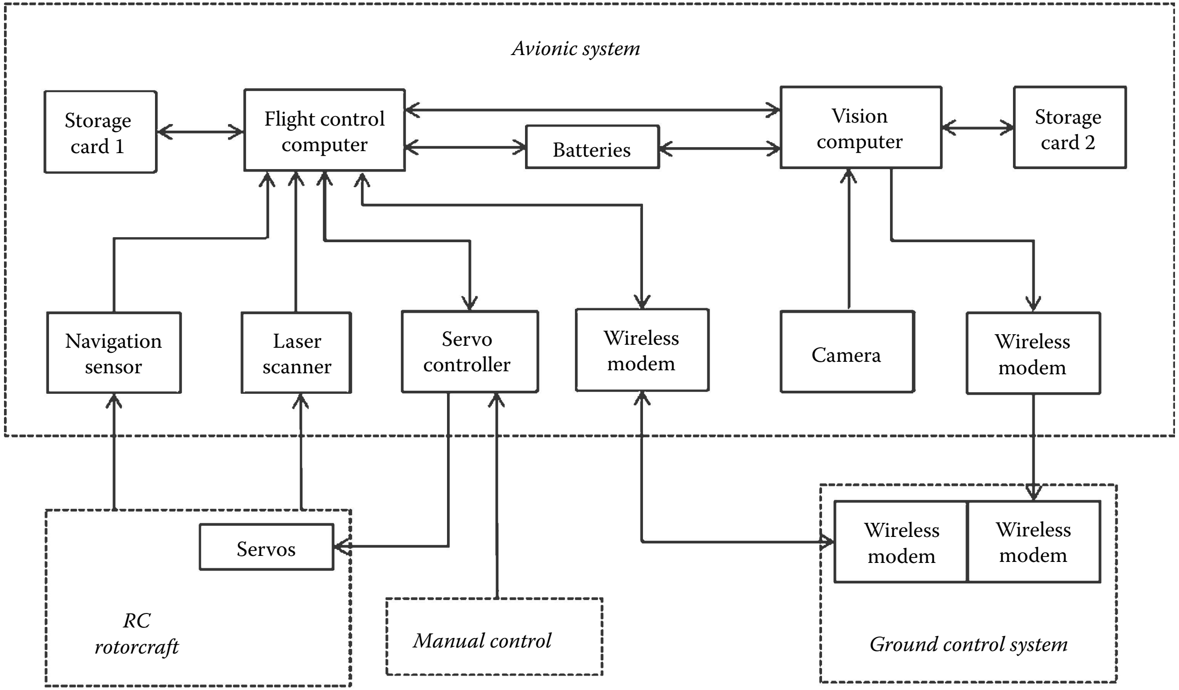

Avionic system
Storage card 1
Flight control computer
Batteries
Vision computer
Storage card 2
Navigation sensor
Laser scanner
Servo controller
Wireless modem
Camera
Wireless modem
Servos
RC rotorcraft
Manual control
Wireless modem
Wireless modem
Ground control system

FIGURE 11.21
Hardware configuration of NUS^{2}T-Lion rotorcraft system.

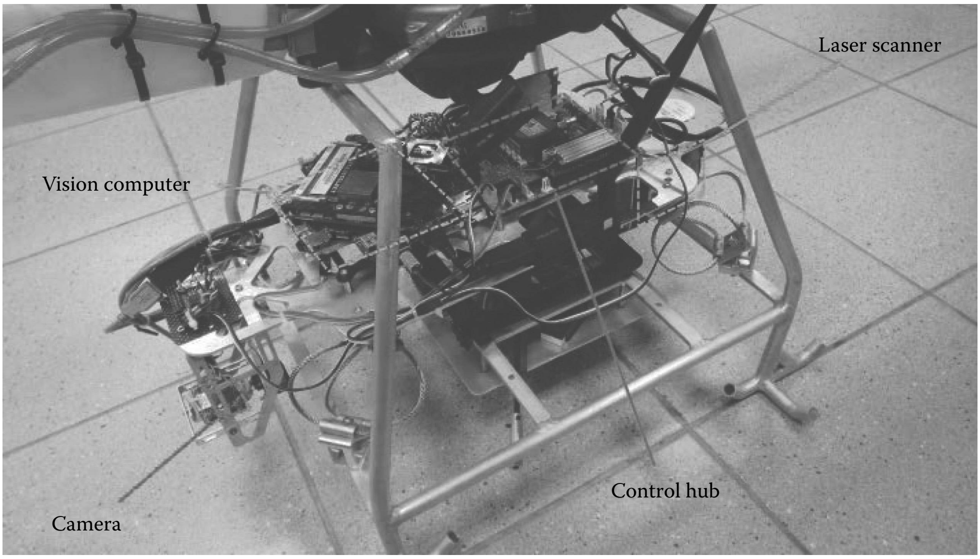


FIGURE 11.22
Onboard avionic system of NUS^{2}T-Lion.

are carefully chosen COTS products that are up to date. Figure 11.22 gives a complete view of the onboard system with the key components indicated. The details and usage of these components are explained as follows.

11.5.1.1 Grabbing Mechanism

For the bare rotorcraft platform, the Thunder Tiger Raptor 90 SE Nitro RC helicopter is adopted in this work. It is a hobby-level single rotor helicopter originally designed for acrobatic flight. As compared with other COTS RC rotorcrafts, such as Turbulence D3 and Observer Twin, Raptor 90 SE provides a reliable structural design and equivalent flight performance at approximately half the price.

However, with the original Raptor 90's nitro engine and nitro fuel tank, the endurance of the UAV can barely reach 8 min with full load avionics. This is not sufficient for practical applications. To overcome this limitation, the original nitro engine is replaced by a gasoline counterpart, which is a product from Zenoah with model number G270RC. With the more efficient gasoline engine, a full-tank Raptor 90 can fly up to 30 min. This greatly widens the range of potential applications for this UAV, and it is especially beneficial to the cargo transportation task.

Unfortunately, this endurance improvement comes with two tradeoffs. First, the vibration of the whole platform intensifies due to the gasoline engine. Second, the ignition magnet inside Zenoah G270RC is so large that its magnetic field can badly affect the onboard sensors. To overcome the vibration issue, wire rope isolators are used to protect the onboard avionics and filter out unwanted high-frequency noises. For the problem of magnetic interference, the final solution boils down to replacing the electromagnetic ignition system inside the engine with a pure electric ignition system. With this modification, the onboard sensors, especially the AHRS, all work as they should.

To cope with the cargo transportation task, there must be a loading mechanism installed on the helicopter. By comparing the solution of a rigid claw-like grabbing mechanism and a long flexible rope hooking mechanism, the former is more precise in picking up the cargo, and the latter can avoid descending the UAV too low to the ship surface where the aerodynamic ground effect becomes an issue. In this work, an innovative design incorporating advantages from both sides has been proposed. The solution is a claw-like grabbing mechanism with very long arms (see Figure 11.23). With this design, the UAV can keep a safe distance from the ship's surface and, at the same time, grab and release the cargo in a precise and reliable way. Another highlight of this design is its omnidirectional feature, meaning no matter in which direction the cargo handle is oriented, it is not necessary for the UAV to adjust its heading to align accordingly. This saves time and minimizes unnecessary maneuvers, which may induce higher risks in autonomous flights. In addition, this design features a self-locking mechanism commonly used in landing gears of hobby-grade fixed-wing planes. The mechanism is enclosed in the rectangular boxes as shown in Figure 11.23 with each box supporting one arm and powered by one servo motor. When the claw fully opens or closes, there is a slider inside the box to lock the position of the servo motor. In this way, the servo motors consume zero power while carrying heavy cargo as the cargo weight is fully supported by the locking mechanism.

A load-sensing mechanism that can differentiate a successful cargo loading from a failure is also installed. This mechanism acts as a safeguard in cases in which the UAV fails to grab the cargo. By knowing that the cargo is not successfully loaded, the UAV can be commanded to descend and grab the cargo again. The detailed design is shown in Figure 11.24, in which four limit switches, which send out electrical signals when pushed down, are installed on the customized landing skid. The baseplate of the claw is rigidly attached to a hollow rectangular plate on its top. The rectangular plate is then resting on the crossover beams of the landing skid via four springs. When the claw is loaded, the rectangular plate compresses the spring and triggers one or more of the limit switches. When the claw is unloaded, the springs push up the rectangular plate to release the limit switches.

In order to retain the UAV x- and y-axis CG balancing, the claw needs to be installed precisely under the UAV CG. In this way, the UAV roll and pitch dynamics will not change

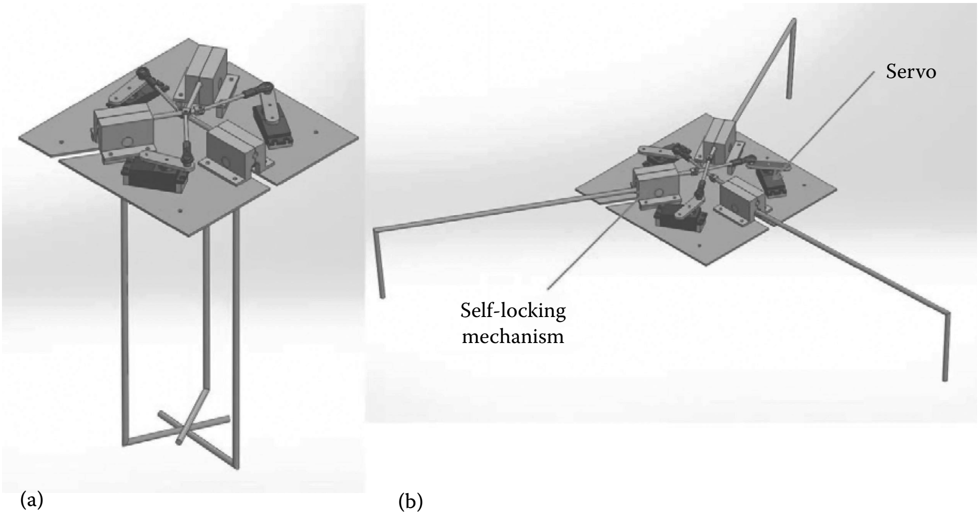


FIGURE 11.23
Grabbing mechanism in closed (a) and open (b) configurations.

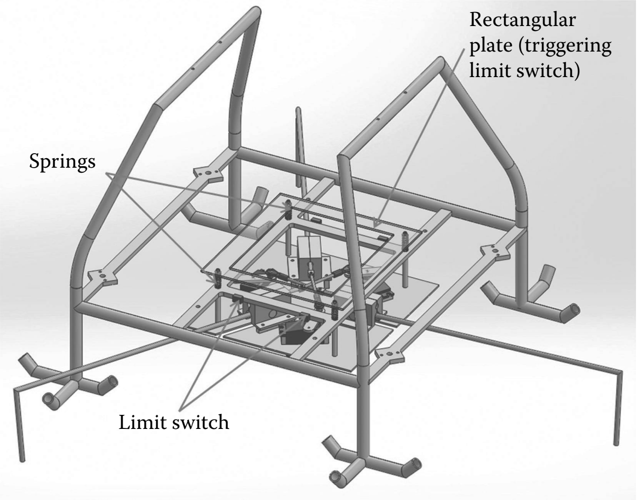


FIGURE 11.24
Grabbing mechanism with load-sensing function.

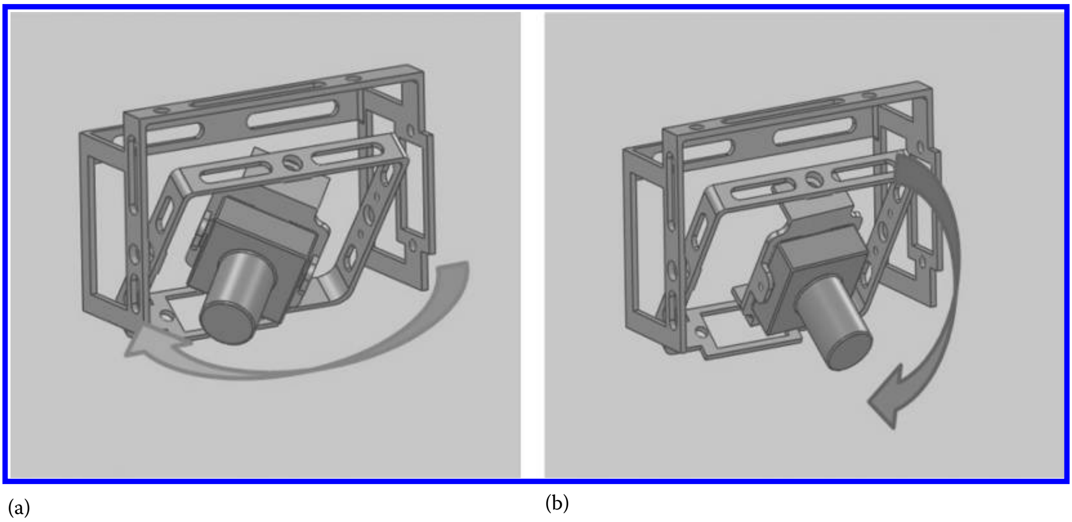
too much after cargo loading; thus the same set of control laws can be used. It also makes sure that controlling the UAV CG to the correct planar position is equivalent to controlling the claw to the correct position, which makes the problem easier.

11.5.1.2 Sensors and Measurement Systems

In addition to the fundamental navigation sensor: IG-500N, another main sensor used onboard the NUS^{2}T-Lion is the mvBlueFOX camera from Matrix Vision. It is a compact industrial CMOS camera, compatible with any computers with USB ports. A superior image quality makes it suitable for both indoor and outdoor applications. Its high-speed USB interface guarantees easy integration without any additional interface board. In this specific cargo transportation application, it is the main guidance sensor for locating the cargo and their unloading points.

By considering the fact that the UAV usually flies forward to search for targets and hovers right above the cargo for loading and unloading, the best position to place the camera is at the nose of the helicopter. In addition, a controlled pan-tilt gimbal (see Figure 11.25) is designed to host the camera sensor so that it always looks vertically downward despite the UAV's rolling and pitching motions. Taking advantage of the camera's wide viewing angle, even when the UAV descends to the lowest altitude for cargo grabbing, the camera can still see the cargo without any problem.

For this cargo transportation application, height measurement from GPS/INS or the barometer is not accurate enough for the UAV to pick up or drop the cargo appropriately. The UAV may even crash on the surface of the cargo platform because of inaccurate height measurement, resulting in catastrophic consequences. Although a vision sensor or 1-D laser range finder may accomplish the task, the former can only be relied on when the visual target is within the field of view, and the latter cannot handle ground surfaces with scattered obstacles. To make the height measurement accurate and consistent, a scanning laser range finder is the ideal choice. The laser scanner code named URG-30LX from Hokuyo is installed in the system. It has a maximum range of 30 m with fine resolution of 50 mm, and it can scan its frontal 270° fan-shaped area with a resolution of 0.25°.



(a) (b)

FIGURE 11.25
Pan-tilt mechanism of the camera: (a) camera pan, (b) camera tilt.

11.5.1.3 Computers

There are two onboard computers in the avionic system: one for the implementation of navigation and control algorithms and the other, more powerful, one dedicated for vision processing. With this dual-computer structure, the vision algorithm can be implemented and tested separately at the development stage, and it is very convenient to upgrade to a more powerful vision computer in the future without modifying the control hardware and software system. It also improves the reliability of the overall system because this structure ensures control stability even when the vision computer malfunctions or encounters run-time errors (it happens more frequently on the vision computer compared to the control counterpart because the vision algorithm usually involves more sophisticated calculations and logics). If it ever happens, the UAV should still fly safely with the control computer alone, and there will be enough time for the human pilot to take over and land the UAV safely.

For the onboard control computer, it collects measurement data from various sensors, performs sensor filtering and fusion, executes flight control law, and outputs control signals to carry out the desired control actions. In addition, it is also responsible for communicating with the GCS as well as data logging. To select a lightweight yet powerful embedded computer for these real-time tasks, the Gumstix Overo Fire embedded computer becomes the final choice.

For the onboard vision computer, it is mainly for implementing image processing algorithms, including color segmentation, object identification, object tracking, and localization. Image processing tasks are usually computationally intensive and hence require powerful processors to run the algorithms in real time. We have chosen the Mastermind computer from Ascending Technologies. It has an Intel Core i7 processor but is small and light enough to be carried by NUS^{2}T-Lion. It also has abundant communication ports to interact with peripheral devices, such as USB cameras and WiFi devices. One UART port is used to communicate with the aforementioned control computer.

11.5.2 Software System

To implement the aforementioned GNC algorithm and to solve the logic problems in completing the UAVGP tasks as a unified process, a robust real-time software system needs to be developed. The following will show the key software concepts of the NUS^{2}T-Lion, which includes the multiple-layer and multiple-thread software structure and mission logic in solving the UAVGP competition tasks.

The software structure of the NUS^{2}T-Lion system is illustrated in Figure 11.26. It consists of three separate software programs, namely the onboard control software, the onboard vision software, and the GCS software. For the onboard control and vision software, they both utilize the multiple-thread framework so that time resources can be precisely distributed among different functional blocks (threads). It is also a more reliable way of implementing real-time UAV software so that the malfunction of an individual thread will not halt the executing of others.

The onboard control software is developed using QNX Neutrino, which provides reliable support for high-precision timer and synchronization operations. Multiple tasks (threads), including operating with hardware devices, such as the navigation sensor, the laser scanner, and the servo control board; implementing the automatic control algorithms; logging in-flight data; and communicating with the GCS and the vision computer, are managed by the MAIN function. With the multiple-thread framework, it is also easy to run different threads with different frequencies. More details about this onboard control software can be found in [62].

Similarly, the onboard vision software is also divided by multiple tasks, namely image capturing from the camera sensor, image processing, data logging, and communication with GCS and the control computer. The operating system utilized on the vision computer is the very popular Ubuntu Linux. It supports abundant hardware drivers, such as USB cameras and WiFi adapters, and software libraries, such as OpenCV, which are very suitable for the development of complicated vision algorithms.

For the GCS software, it runs on a laptop with a Windows 7 system. Such a commercial operating system provides strong support for the development of user interfaces. Visual C++ is employed to develop and debug the GCS software. By utilizing the MFC library, the global shared data are hosted in a document class, in which a variety of functions for data operating and visiting are integrated. Although this document class is the kernel of the software program, there are also the communication thread and other threads to control the multiple views at the foreground user interface. The communication thread receives and sends data to the UAV onboard system through a WiFi link, and the multiple displaying views periodically visit the contents of the document and update their respective displaying of new data received.

As the UAV cargo transportation application is mission-oriented, the sequence and logics of the whole task operation are rather important. It is implemented in the CTL thread of the onboard control software. The overall *Mission Logic* is illustrated in Figure 11.27. It consists of six sequential tasks, namely takeoff, navigate to ship, vision initialization, transporting cargo, return home, and landing. Because the mission is time-constrained, a timer interrupt is also implemented in the software. The timer interrupt will trigger the return home task once the predefined maximum mission duration runs out. The details of each task are discussed as follows:

1. The takeoff task will be triggered once the onboard software program receives the "Mission Start" command from the GCS. In this stage, the program lets the helicopter warms up its engine by issuing a small and constant value to the throttle

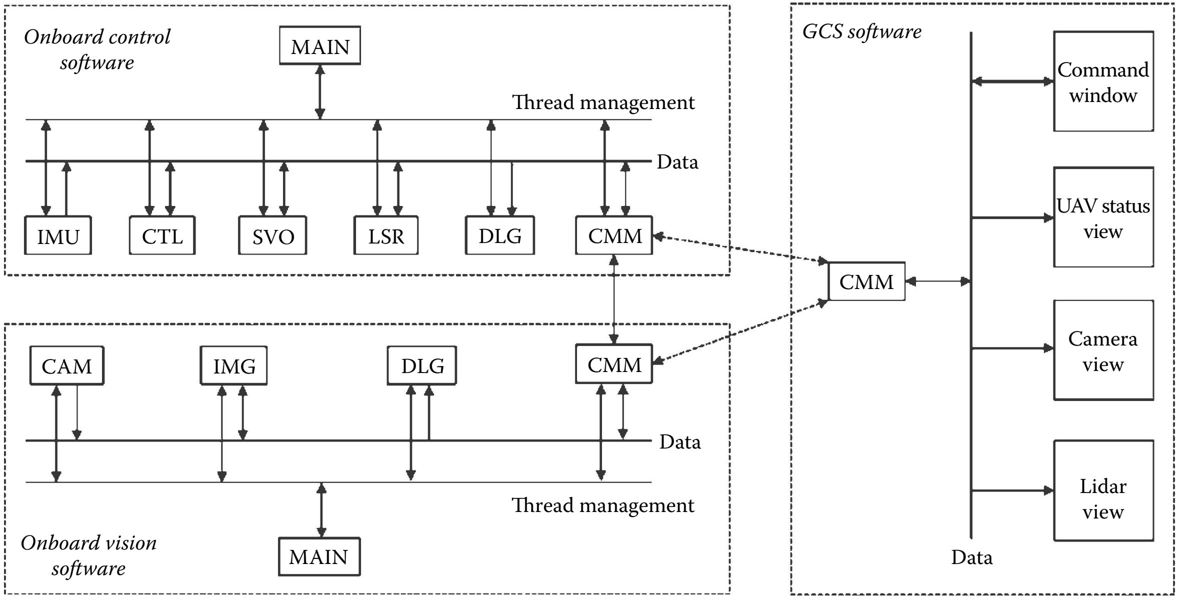

Onboard control software
MAIN
Thread management
Data
IMU
CTL
SVO
LSR
DLG
CMM
GCS software
Command window
UAV status view
CMM
Camera view
Lidar view
Data
CAM
IMG
DLG
CMM
Data
Thread management
Onboard vision software
MAIN

FIGURE 11.26
Software structure of NUS^{2}T-Lion. CAM, image capture from camera sensor; CMM, communication; CTL, control law implementation; DLG, data logging; IMG, image processing; IMU, measurement reading from GPS/INS; LSR, laser measurement and processing; MAIN, main program, task management; SVO, servo driving and reading.

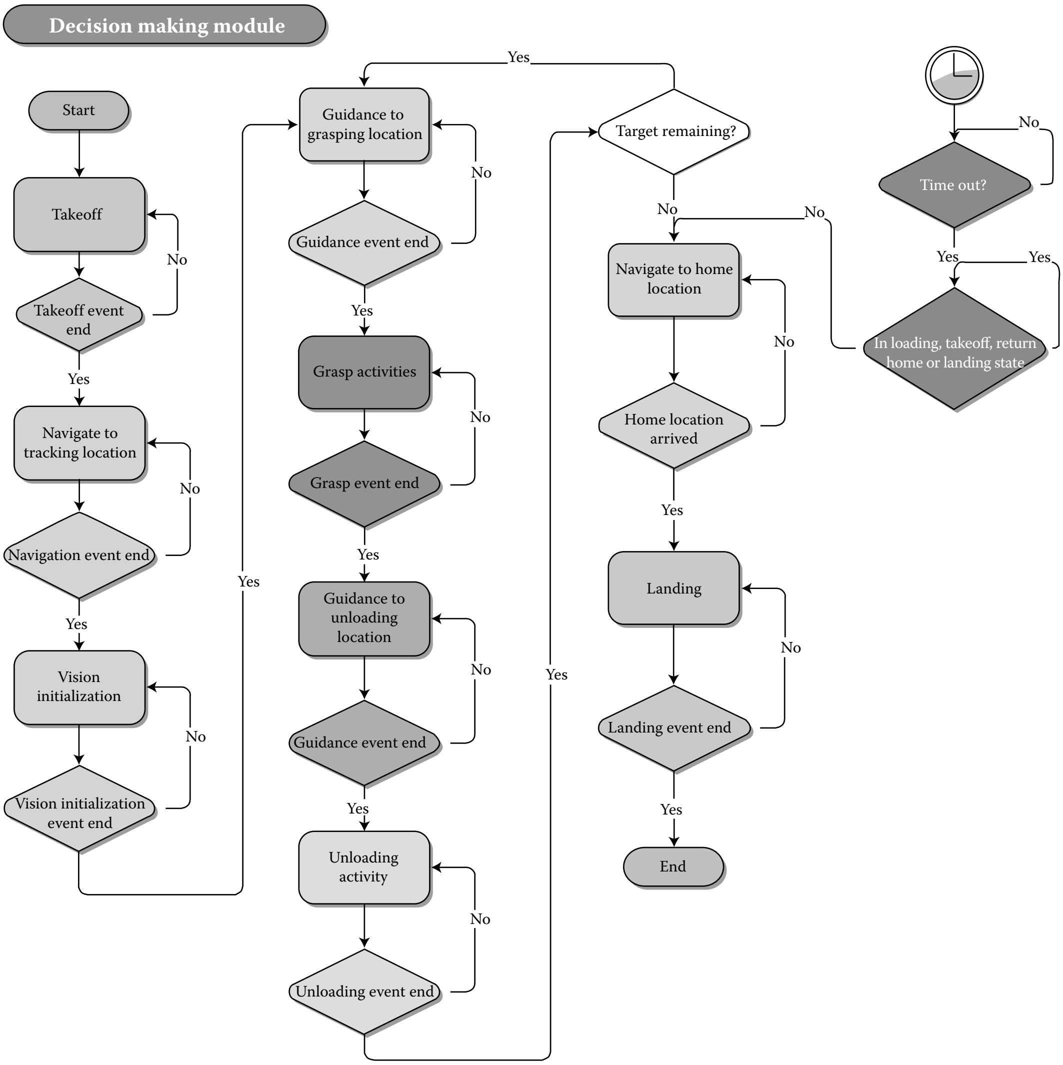

Decision making module
Start
Takeoff
No
Takeoff event end
Yes
Navigate to tracking location
No
Navigation event end
Yes
Vision initialization
No
Vision initialization event end
Yes
Guidance to grasping location
No
Guidance event end
Yes
Grasp activities
No
Grasp event end
Yes
Guidance to unloading location
No
Guidance event end
Yes
Unloading activity
No
Unloading event end
Yes
Yes
Target remaining?
No
No
Navigate to home location
No
Home location arrived
Yes
Landing
No
Landing event end
Yes
End
No
Time out?
Yes
Yes
In loading, takeoff, return home or landing state

FIGURE 11.27
Mission logics.

channel. After a while, the throttle channel control signal will be increased gradually until the engine enters the governor mode (main blades will now be controlled at a predefined rotational speed of 1750 rpm). After that, the program will slowly increase the control signal of the collective pitch channel so that the lift force increases. Once the collective pitch signal hits its trimming value for the hovering condition, the program will ask the reference generation function to issue a "Going Up" trajectory. At the end of the trajectory, the program throws a "Takeoff Event End" signal.

2. The software program now enters the "Navigate to Ship" mode. In this stage, the program collects the position and velocity information from the GPS/INS system on the ship. A relative path to the ship with continuous relative position, relative velocity, and relative acceleration references will be generated. The flight controller

will continuously ask the helicopter to track this path so that the helicopter can catch up with the ship and have the same position and velocity profiles as the ship at steady state. Once the helicopter catches up with the ship, the software will throw a "Navigation Event End" signal. Note that this decision is made based on GPS/INS information. Physically the UAV may not be hovering so precisely above the center of the two ships.

3. In the "Vision Initialization," the vision system will first check whether it can detect two ships. If only one ship or part of a ship has been detected, the vision system will guide the helicopter to move toward one of the detected circles. In this way, there will be very high probability to see the other ship by taking advantage of the onboard camera's wide viewing angle. Once both ships are successfully detected, the software will be scheduled to the "Transporting Cargo" mode.
4. The "Transporting Cargo" task is the most sophisticated part of the mission. In this stage, the UAV will choose the cargo and fly to a position right above it. When the UAV horizontal position to the target cargo enters a small threshold, its height reference will be gradually adjusted down to an appropriate value so that the mechanical claw can grasp the bucket handle. Once the mechanical claw is closed, the UAV will be commanded to fly upward quickly so that the limit switch sensors mounted under the claw platform can sense whether the cargo weight has been successfully loaded. If it deduces that cargo has not been grasped successfully, the UAV will be commanded to go down again for another try. The above procedure will be repeated until the limit switch system detects a successful cargo loading. After that, the helicopter will be commanded to move to the unloading point. For the unloading task, the UAV has a similar procedure to check whether the cargo has been successfully released. If the detection is false, the UAV will quickly close and open its claw to try another release. For failsafe, when the vision system loses the cargo target for more than 10 sec during the grasping stage, the software will issue a "Going Up" command so that the vision system can have a wider view, which leads to higher chance of retrieving the target. Once the vision system retrieves the target, the UAV will be commanded to go down and try grasping the cargo again. There is a counter to record how many cargos remain to be transported. Once the counter hits zero, the program will jump out the current mode and enter the return home mode.
5. When the helicopter has finished all its transportation tasks or the maximum mission time runs out, the "Return Home" task will be triggered. The software will generate a reference trajectory ending at a predefined height and with the UAV's planar position equal to the initial takeoff point. The UAV will then follow this trajectory back to the home location.
6. The landing will be triggered as the helicopter flies right above its home location. The procedure for the landing task is similar to the takeoff task. The software asks the flight controller to regulate the helicopter moving downward with a constant speed at 0.5 m/s (if height is greater than 5 m) or 0.2 m/s (if height is less than 5 m). Once the UAV landing gear approaches the ground (within 8 cm), the control signal to the throttle channel will jump to a minimum value so that the engine shuts down.

11.5.3 Experimental Results

In preparation for the UAVGP competition, numerous flight tests have been carried out to verify the overall solution feasibility and to tune for the optimal performance. Figures 11.28

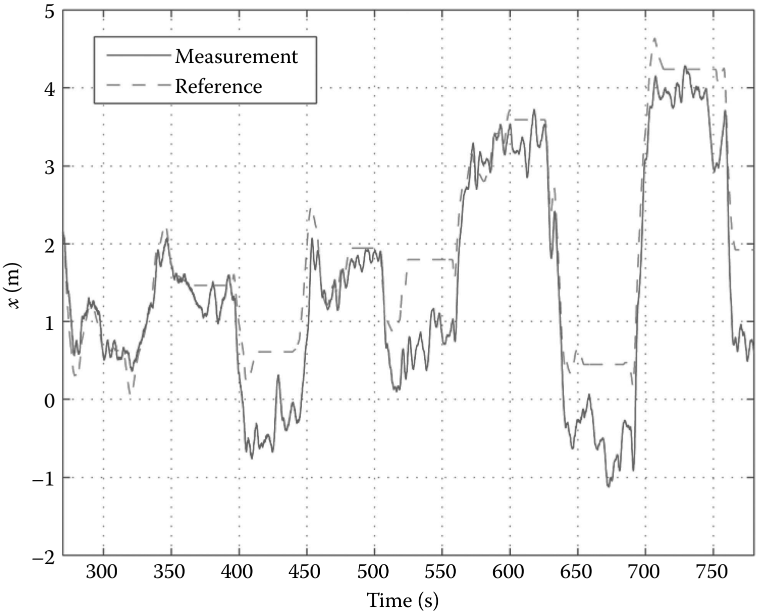


FIGURE 11.28
UAV position response in the ship-frame *x*-axis.

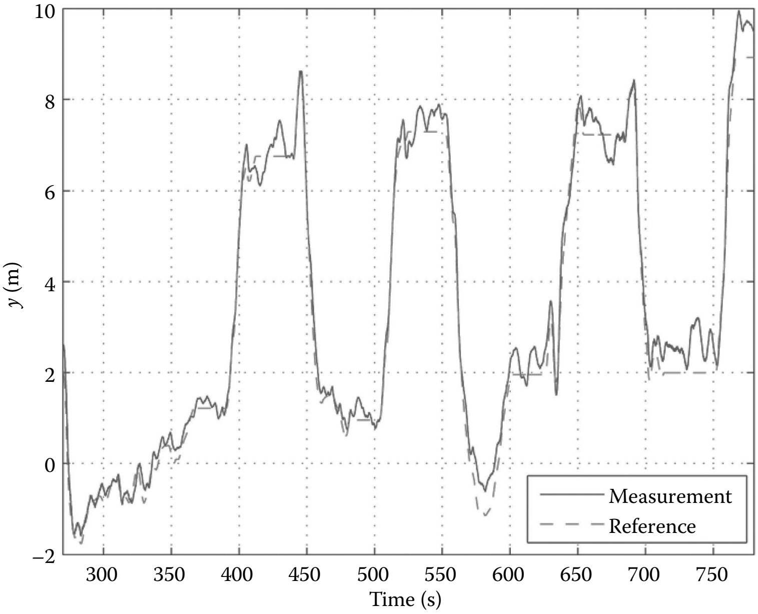


FIGURE 11.29
UAV position response in the ship-frame *y*-axis.

through 11.30 show the position data logged in one of the flight tests. As the raw data is obtained by GPS/INS and then converted to the ship frame, it may not be the exact truth. However, it generally shows the control performance and roughly indicates whether the UAV is doing the correct movement. In Figure 11.28, the x-signal becomes larger progressively because the UAV is moving from the first bucket to the fourth bucket. It always comes back to a position around zero because the reference path is defined in a way that the onboard camera has the best view of the two ships before every loading or unloading dive. In Figure 11.29, the y-position signal goes back and forth, indicating alternative movements between the two ships. In Figure 11.30, it is clear to see all the diving motions of the UAV. The UAV will stay at a very low altitude with variable duration depending on how many loading or unloading trials have been performed before the successful one.

With this kind of performance, the NUS^{2}T-Lion has successfully accomplished the competition tasks in the UAVGP rotary-wing category. A final score of 1127.56 with 472.44 from the preliminary contest and 655.13 from the finals has pushed the team into second position in the overall Grand Prix. It should be highlighted that 655.13 was the highest score among the finalists. Moreover, unlike the preliminary contest, the final round of competition requires the UAV to carry out the cargo transportation task with the two ships moving. This demands better robustness and higher intelligence from the participants' UAV systems, and it is indeed the strongest point of the GNC solution proposed. The final competition has been recorded in video format and uploaded to [70] and [71] for the English and Chinese versions, respectively.

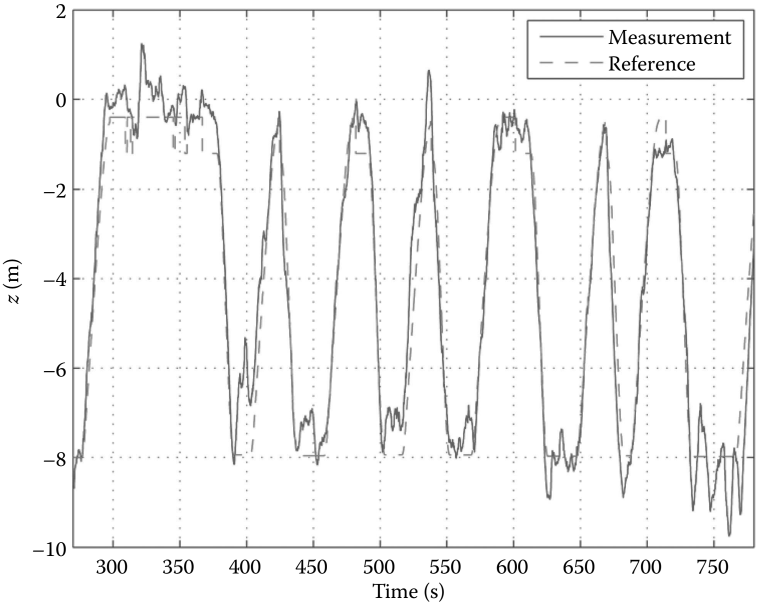


FIGURE 11.30
UAV position response in the NED-frame z-axis.

11.6 Conclusion

In this chapter, we have presented the mechatronics design of UAV systems, including hardware systems and software systems. The modular design approach and computer-aided hardware integration were highlighted in the hardware system construction. The multitask embedded software design has been presented in the software system development. After that, we addressed two case studies including a coaxial rotorcraft GremLion developed for the DARPA UAVForge challenge and a cargo transportation system used in UAVGP 2013. The experimental results show that the proposed methodology is efficient and effective. The constructed UAV systems can be used as excellent platforms for future research development.

In summary, system thinking is the critical part in mechatronics design of unmanned aircraft systems. How well we understand the mission requirements and the characteristics of possible components of the unmanned system plays a crucial role in mechatronics design. A balance between the system performance and the total cost in development, operation and maintenance needs to be achieved to fulfill a good mechatronics design. The proposed systematic design methodology will be able to facilitate in this process. Although we are faced with many obstacles and challenges in these research areas, component development and system integration, including hardware, software, and algorithms will continuously be improved to advanced levels. With the development in avionics, GCS, and mission algorithms, other unmanned systems will also be benefited by mechatronics design with systems thinking.

References

1. G. Cai, B. M. Chen, and T. H. Lee, *Unmanned Rotorcraft Systems*. New York: Springer, 2011.
2. Z. Sarris and S. Atlas, "Survey of UAV applications in civil markets," in *Proceedings of the 9th Mediterranean Conference on Control and Automation*, vol. WA2-A, Dubrovnik, Croatia, 2001, pp. 1–11.
3. B. Enderle, "Commercial applications of UAV's in Japanese agriculture," in *Proceedings of the AIAA 1st UAV Conference*, no. AIAA-2002-3400, Portsmouth, VA, May 20–23, 2002.
4. B. Ludington, E. Johnson, and G. Vachtsevanos, "Augmenting UAV autonomy," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 63–71, 2006.
5. M. E. Campbell and W. W. Whitacre, "Cooperative tracking using vision measurements on seascan UAVs," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 613–626, 2007.
6. J. Keller, D. Thakur, V. Dobrokhodov, K. Jones, M. Pivtoraiko, J. Gallier, I. Kaminer, and V. Kumar, "A computationally efficient approach to trajectory management for coordinated aerial surveillance," *Unmanned Systems*, vol. 1, pp. 59–74, 2013.
7. A. N. Kopeikin, S. S. Ponda, L. B. Johnson, and J. P. How, "Dynamic mission planning for communication control in multiple unmanned aircraft teams," *Unmanned Systems*, vol. 1, no. 1, pp. 41–58, 2013.
8. K. Kang and J. V. R. Prasad, "Development and flight test evaluations of an autonomous obstacle avoidance system for a rotary-wing UAV," *Unmanned Systems*, vol. 1, no. 1, pp. 3–19, 2013.
9. F. Lin, B. M. Chen, and T. H. Lee, "Robust vision-based target tracking control system for an unmanned helicopter using feature fusion," in *Proceedings of the 11th IAPR Conference on Machine Vision Applications*, Yokohama, Japan, 2009, pp. 398–401.
10. T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Synthesis of control protocols for autonomous systems," *Unmanned Systems*, vol. 1, pp. 75–119, 2013.

11. M. J. Adriaans, R. C. Cribbs, B. T. Ingram, and B. P. Gupta, “A160 Hummingbird unmanned air vehicle development program,” in *Proceedings of the AHS International Specialists’ Meeting—Unmanned Rotorcraft: Design, Control and Testing*, Chandler, AZ, 2007.
12. G. W. Cai, B. M. Chen, K. Peng, M. Dong, and T. H. Lee, “Modeling and control system design for a UAV helicopter,” in *Proceedings of the 14th Mediterranean Conference on Control and Automation*, vol. WM1-4, Ancona, Italy, 2006, pp. 1–6.
13. V. Gavrilets, A. Shterenberg, M. A. Dahleh, and E. Feron, “Avionics system for a small unmanned helicopter performing aggressive maneuvers,” in *Proceedings of the 19th Digital Avionics Systems Conferences*, Philadelphia, PA, 2000, pp. 1E2/1–1E2/7.
14. J. M. Roberts, P. Corke, and G. Buskey, “Low-cost flight control system for a small autonomous helicopter,” in *Proceedings of the 2002 Australian Conference on Robotics and Automation*, Auckland, New Zealand, 2002, pp. 546–551.
15. K. Sprague, V. Gavrilets, D. Dugail, B. Mettler, E. Feron, and I. Martinos, “Design and applications of an avionic system for a miniature acrobatic helicopter,” in *Proceedings of the 20th Digital Avionics Systems Conferences*, Daytona Beach, FL, 2001, pp. 3C5/1–3C5/10.
16. K. M. Peng, G. W. Cai, B. M. Chen, M. B. Dong, K. Y. Lum, and T. H. Lee, “Design and implementation of an autonomous flight control law for a UAV helicopter,” *Automatica*, vol. 45, no. 10, pp. 2333–2338, 2009.
17. A. Karimoddini, H. Lin, B. M. Chen, and T. H. Lee, “Hybrid 3-d formation control for unmanned helicopters,” *Automatica*, vol. 49, no. 2, pp. 424–433, 2013.
18. S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained MAV,” in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 20–25.
19. F. Wang, J. Q. Cui, B. M. Chen, and T. H. Lee, “A comprehensive UAV indoor navigation system based on vision optical flow and laser fastSLAM,” *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1889–1900, 2013.
20. J. Kim and S. Sukkarieh, “Autonomous airborne navigation in unknown terrain environments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 1031–1045, 2004.
21. H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping (SLAM): Part I,” *IEEE Robotics & Automation Magazine*, vol. 13, pp. 99–110, 2006.
22. J. H. Kim, J. Lyou, and H. K. Kwak, “Vision coupled GPS/INS scheme for helicopter navigation,” *Journal of Mechanical Science and Technology*, vol. 24, no. 2, pp. 489–496, 2010.
23. A. Martinelli, “Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.
24. P. Doherty, F. Heintz, and J. Kvarnstrom, “High-level mission specification and planning for collaborative unmanned aircraft systems using delegation,” *Unmanned Systems*, vol. 1, no. 1, pp. 75–119, 2013.
25. J. Hu, J. Xu, and L. Xie, “Cooperative search and exploration in robotic networks,” *Unmanned Systems*, vol. 1, pp. 121–142, 2013.
26. U.S. Government, Department of Defense, U.S. Military, “Unmanned systems integrated roadmap fy 2013–2038,” Department of Defense, Tech. Rep., December 2013.
27. A. K. Kundu, *Aircraft Design*. The Edinburgh Building, Cambridge, UK: Cambridge University Press, 2010.
28. R. Austin, *Unmanned Aircraft Systems*. UK: Wiley, Chichester, West Sussex, 2010.
29. A small list of IMU/INS/INU. [Online]. Available: http://damien.douxchamps.net/research/imu/.
30. P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, London, 2008.
31. H. Chao, C. Coopmans, L. Di, and Y. Q. Chen, “A comparative evaluation of low-cost IMUs for unmanned autonomous systems,” in *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Salt Lake City, UT, September 2010, pp. 211–216.

32. B. Mettler, M. Tischler, and T. Kanade, “Circular error probable (CEP),” USA: Air Force Operational Test and Evaluation Center, Tech. Rep., 1987.
33. G. C. Guenther, A. G. Cunningham, P. E. LaRocque, and D. J. Reid, “Meeting the accuracy challenge in airborne lidar bathymetry,” in *Proceedings of EARSeL-SIG-Workshop LIDAR*, June 2000, pp. 1–27.
34. Phoneix aerial UAV lidar. [Online]. Available: http://www.phoenix-aerial.com/.
35. D. Wood and M. Bishop, “A novel approach to 3D laser scanning,” in *Proceedings of Australasian Conference on Robotics and Automation*, New Zealand, 2012.
36. H. Baltzakis, A. Argyros, and P. Trahanias, “Fusion of laser and visual data for robot motion planning and collision avoidance,” *Machine Vision and Applications*, vol. 15, no. 2, pp. 92–100, 2003.
37. O. Amidi, T. Kanade, and R. Miller, “Vision-based autonomous helicopter research at Carnegie Mellon robotics institute 1991–1997,” in *Proceedings of American Helicopter Society International Conference*, Gifu, Japan, 1998, pp. 1–12, paper no. T73.
38. N. Guenard, T. Hamel, and R. Mahony, “A practical visual servo control for an unmanned aerial vehicle,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, 2008.
39. E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, “Real-time vision-based relative aircraft navigation,” *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 707–738, 2007.
40. S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, “Combined optic-flow and stereo-based navigation of urban canyons for a UAV,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3309–3316.
41. O. Shakernia, Y. Ma, T. J. Koo, and S. Sastry, “Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control,” *Asian Journal of Control*, vol. 1, pp. 128–204, 1999.
42. C. S. Sharp, O. Shakernia, and S. S. Sastry, “A vision system for landing an unmanned aerial vehicle,” in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea, 2001, pp. 1720–1727.
43. J. Kim and S. Sukkarieh, “Slam aided GPS/INS navigation in GPS denied and unknown environments,” in *The 2004 International Symposium on GNSS/GPS*, Sydney, Australia, 2004.
44. M. Meingast, C. Geyer, and S. Sastry, “Vision based terrain recovery for landing unmanned aerial vehicles,” in *Proceedings of IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004, pp. 1670–1675.
45. K. Litomisky, “Consumer RGB-D cameras and their applications,” Technical Report, University of California, 2012.
46. P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments,” in *Proceedings of the 12th International Symposium on Experimental Robotics (ISER)*, 2010.
47. P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
48. H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, “Interactive 3D modeling of indoor environments with a consumer depth camera,” in *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.
49. A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual odometry and mapping for autonomous flight using an RGB-D camera,” in *Proceedings of the International Symposium on Robotics Research*, 2011.
50. E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Proceedings of the 9th European Conference on Computer Vision*, 2006.
51. N. Fioraio and K. Konolige, “Realtime visual and point cloud SLAM,” in *Proceedings of the RGB-D Workshop, Robotics Science and Systems*, 2011.
52. F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D SLAM system,” in *ICRA*, IEEE, 2012, pp. 1691–1696.

53. G. Cai, F. Lin, B. M. Chen, and T. H. Lee, "Systematic design methodology and construction of UAV helicopters," *Mechatronics*, vol. 18, no. 10, pp. 545–558, 2008.
54. S. G. Fowers, D. J. Lee, B. J. Tippetts, K. D. Lillywhite, A. W. Dennis, and J. K. Archibald, "Vision aided stabilization and the development of a quad- rotor micro UAV," in *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL, 2007, pp. 143–148.
55. R. He, A. Bachrach, M. Achtelik, A. Geramifard, D. Gurdan, S. Prentice, J. Stumpf, and N. Roy, "On the design and use of a micro air vehicle to track and avoid adversaries," *The International Journal of Robotics Research*, vol. 29, pp. 529–546, 2010.
56. M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *The First Symposium on Indoor Flight Issues 2009*, Mayagez, Spanish, 2009.
57. S. K. Phang, J. J. Ong, R. T. C. Yeo, B. M. Chen, and T. H. Lee, "Autonomous mini-UAV for indoor flight with embedded on-board vision processing as navigation system," in *Proceedings of the IEEE R8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Irkutsk Listvyanka, Russia, 2010.
58. Wikipedia. Real-time operating system. [Online]. Available: http://en.wikipedia.org/wiki/Real-time operating system.
59. Qnx neutrio rtos. [Online]. Available: http://www.qnx.com/.
60. Vxworks rtos. [Online]. Available: http://www.windriver.com/products/vxworks/.
61. Rt linux rtos. [Online]. Available: http://www.rtlinuxfree.com/.
62. M. Dong, B. M. Chen, G. Cai, and K. Peng, "Development of a real-time onboard and ground station software system for a UAV helicopter," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 8, pp. 933–955, 2007.
63. Visual C++ developer center. [Online]. Available: http://msdn.microsoft.com/en-us/visualc/default.aspx.
64. F. Lin, Z. Y. Ang, F. Wang, B. M. Chen, T. H. Lee, B. Q. Yang, M. B. Dong et al., "Development of an unmanned coaxial rotorcraft for the DARPA UAVForge challenge," *Unmanned Systems*, vol. 1, pp. 211–245, 2013.
65. A. Koehl, H. Rafaralahy, M. Boutayeb, and B. Martinez, "Aerodynamic modelling and experimental identification of a coaxial-rotor UAV," *Journal of Intelligent & Robotic Systems*, vol. 68, pp. 53–68, 2012.
66. B. Mettler, M. Tischler, and T. Kanade, "System identification of a model-scale helicopter," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-00-03, January 2000.
67. C. Harris, *Shock and Vibration Handbook*, 4th ed. New York: McGraw-Hill, 1996.
68. D. J. Kruglinski, *Inside Visual C++*, 4th ed. Washington, DC: Microsoft Press, 1995.
69. P. Liu, X. Dong, B. M. Chen, and T. H. Lee, "Development of an enhanced ground control system for unmanned aerial vehicles," in *Proceedings of the IASTED International Conference on Engineering and Applied Science*, Colombo, Sri Lanka, December 2013, pp. 136–143.
70. Unmanned Aircraft Systems Group, National University of Singapore, "The video of the final round competition of UAVGP (English version)." Available: http://www.youtube.com/watch?v=Ex0jgtX2ZrY, 2013.
71. Unmanned Aircraft Systems Group, National University of Singapore, "The video of the final round competition of UAVGP (Chinese version)." Available: http://v.youku.com/vshow/id XNjI5MTM2MDI0.html, 2013.

12

Self-Powered and Bio-Inspired Dynamic Systems

Farbod Khoshnoud and David J. Dell

CONTENTS

12.1 Introduction 454
12.2 Energy Harvesting 455
    12.2.1 Energy Conversion Mechanisms 459
12.3 Self-Powered Dynamic Systems 462
    12.3.1 Concept of Self-Powered Dynamic Systems 462
    12.3.2 Theory of Self-Powered Systems 463
    12.3.3 Renewable Energy for Dynamic Systems 466
        12.3.3.1 Power System 468
    12.3.4 Human-Powered Systems 470
12.4 Bio-Inspired Dynamic Systems 471
    12.4.1 Piezoelecteric Energy Harvesting from Aeroelastic Vibrations 473
    12.4.2 Fish Schooling Inspired Vertical Axis Wind Turbine Farm 477
    12.4.3 Bio-Inspired Self-Propelled Vehicle 478
    12.4.4 Bio-Inspired Flapping Wing Flying Robots 481
    12.4.5 Bio-Inspired Flight Control System 488
        12.4.5.1 Formation Flight of Birds 490
    12.4.6 Uncertainty Quantification 491
12.5 Conclusions 494
References 494

SUMMARY This chapter presents some case studies of state-of-the-art mechatronic systems. The concept of self-powered dynamic systems is discussed and is defined as a system powered by its own excessive kinetic energy, renewable energy, or a combination of both. The particular area of work is the concept of fully or partially self-powered systems or subsystems requiring zero or reduced external energy inputs. Self-powered sensors and actuators, self-powered controls and solar-powered vehicles are explored in association with self-powered dynamic systems. Energy harvesting techniques are addressed in regards to the application to self-powered dynamic systems. Bio-inspired engineering approaches are investigated in the design of efficient dynamic systems with particular reference to energy efficiency and aerial robotics.

12.1 Introduction

Design of a fully or partially self-powered device that requires zero or reduced external energy input can be achieved by employing energy harvesting techniques. A self-powered dynamic system can benefit from energy harvesting mechanisms with which particular applications are associated with self-powered sensors, actuators, and control systems [1–6]. The self-powered dynamic system can harvest its own excessive kinetic energy, renewable energy, or a combination of both [1]. The power conversion for mechanical kinetic energy to electrical energy can be realized through piezoelectric, electromagnetic, or electrostatic energy conversion mechanisms [7].

Self-powered sensors can use electrostatic, piezoelectric, and electromagnetic energy conversion schemes to provide the required power for sensor operations [8–16]. A self-powered sensor does not require maintenance or replacement of its battery at the end of the battery life. This is particularly beneficial in condition monitoring and remote sensing in hostile or inaccessible environments. An energy harvester can extract the maximum power from ambient vibration or machinery/structural–induced vibration when the excitation frequency matches the natural frequency of the energy harvester. Various mechanisms are designed to tune the natural frequency of the harvester with the excitation frequency in order to maximize the generated power [17–19]. This technology is particularly suitable for ambient energy harvesting, which is a random, multifrequency, and time-varying source [20,21]. For random and broadband excitation frequencies, a higher damping characteristic is preferred in the absence of tuneable frequency technology [22,23]. Energy harvesters can be designed based on the operational frequency range of systems. For instance, low-frequency energy harvesters can generate sufficient power for small electronic components using electromagnetic and piezoelectric mechanisms [24–26].

Regenerative actuators can harvest energy from vibration and use that energy as a power source for their own operation or for powering other electrical instruments [27–31]. For instance the generated power can be used for controlling the source of the vibration itself as a self-powered system or can generate electrical energy to be stored in the system [32–36]. For instance, if such a system is used as a regenerative system to harvest the vibration energy from a road vehicle suspension, it can potentially generate power in the range of tens of W to thousands of W [37–44]. The generated power can be utilized in a vehicle suspension system for controlling the vibration of the vehicle. A self-powered vibration control system converts unwanted vibration energy in the system to electrical energy and uses the same electrical output to control the excessive vibration [45–51].

Energy harvesting in water distribution systems can employ the hydraulic energy, thermal energy, and kinetic energy in the water pressure fluctuation to power wireless sensors [52,53]. Various applications for energy harvesting devices are found. In addition to the examples mentioned above, energy harvesting devices embedded in animal bodies use the kinetic energy of the system as a power source. For instance, a piezoelectric-based shoe sole or clothing fabric can generate electricity from human motion [54].

Inspiration from nature allows the design of more efficient engineering systems. Biological systems and animals are products of nature that have evolved to better respond to life activities. For instance, studies carried out on the mechanics of aquatic animals by biologists have shown that even a dead trout can swim upstream by extracting energy and generating thrust-producing body deflections and that whale flukes are capable of absorbing energy from surface waves and creating thrust whether the whale is alive or dead [55–58]. A fish is capable of utilizing energy in upstream vortices to enhance its locomotion performance and efficiency. It was demonstrated that "through passive body deformations and vorticity control, a fish could even propel itself forward inside a Kármán vortex street without energy expense" [59]. This is the motivation for employing biomimetics to improve engineering systems design. For instance, a piezoelectric energy harvester in a bio-inspired array of harvesters (when placing piezoelectric cantilevers next to each other in configurations analogous to a group of fish swimming next to each other) can generate more power compared to a piezoelectric energy harvester in isolation [60–63]. Vertical axis wind turbines designed on the basis of fish schooling shows an order of magnitude improvement in the power density of the wind farms [64,65].

The subsequent sections discuss the concept of self-powered dynamic systems and applications and biomimetics in improving the design of engineering systems. Concepts of sensing and actuation are covered in Chapter 6.

12.2 Energy Harvesting

Energy harvesting techniques exploit various sources of energy, such as solar power, thermal energy, wind energy, and kinetic energy, for energy scavenging. Specifically, kinetic energy as a source of energy for energy harvesting is explored here in this section. In order to explain the concept of energy harvesting from kinetic energy, the dynamic system in Figure 12.1 is considered.

The equation of motion of the mass-spring-damper system in Figure 12.1 is given by [66,67]

$$m\ddot{x} + c(\dot{x} - \dot{y}) + k(x - y) = 0 \tag{12.1}$$

where y denotes the base excitation, and x is the mass displacement, k is the spring stiffness, and c is the damping constant. If $z = x - y$ is the displacement of the mass relative to the base then Equation 12.1 becomes

$$m\ddot{z} + c\dot{z} + kz = -m\ddot{y} \tag{12.2}$$

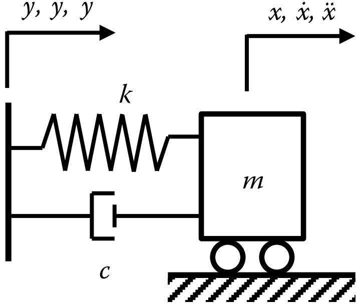


FIGURE 12.1
Mass spring damper system with base excitation.

For a harmonic input displacement excitation of $y = Y\sin\omega t$, the input acceleration excitation is obtained from the second derivative of the displacement with respect to time.

$$\ddot{y} = -m\omega^2 Y \sin\omega t$$

Therefore Equation 12.2 can be written as

$$m\ddot{z} + c\dot{z} + kz = m\omega^2 Y \sin\omega t \tag{12.3}$$

If the steady-state solution for relative displacement is considered as

$$z = Z\sin(\omega t - \phi) \tag{12.4}$$

From Equations 12.3 and 12.4, the amplitude of the relative displacement can be obtained as

$$Z = \frac{m\omega^2 Y}{\sqrt{(k - \omega^2 m)^2 + c^2\omega^2}} \tag{12.5}$$

The energy in the mass-spring-damper system includes kinetic energy of the moving mass, elastic energy of the spring, and the energy loss in the damper. Part of the kinetic energy of the mass-spring-damper system is converted to heat energy in the damper, which is considered to be wasted energy. An energy harvester system can be designed to recover this wasted energy or convert excessive kinetic energy of the system into useful energy [1–6]. Therefore, the dissipated energy from the damper or the excessive kinetic energy can be harvested through a regenerative mechanism. The potential amount of power that can be harvested if the wasted energy in the damper is recovered by a regenerative mechanism can be obtained as

$$P = c\dot{z} \times \dot{z} \tag{12.6}$$

From Equation 12.4, $\dot{z} = \omega Z\cos(\omega t - \phi)$, and therefore, harvested, the power is determined as

$$P = c\omega^2 Z^2 \cos^2(\omega t - \phi) \tag{12.7}$$

The amount of energy in one cycle of oscillation of the mass-spring-damper system with a period of $\tau = 2\pi/\omega$ can be obtained by integrating the power in Equation 12.7 with respect to time as follows:

$$E = c\omega^2 Z^2 \int_0^{2\pi/\omega} \cos^2(\omega t - \phi)\,\mathrm{dt} = \pi c\omega Z^2 \tag{12.8}$$

The average power can be calculated by dividing the amount of energy in one oscillation cycle (Equation 12.8) by the period of oscillation as

$$P = E/\tau = c\omega^2 Z^2/2 \tag{12.9}$$

From Equations 12.5 and 12.9, the power is obtained as

$$P = \frac{cm^2\omega^6Y^2}{2\left((k-\omega^2m)^2+c^2\omega^2\right)} \tag{12.10}$$

It is now desired to obtain a dimensionless power [22] value for generalization of the power value. From Equation 12.5, the amplitude of the excitation displacement can be presented as

$$Y = \frac{Z\sqrt{(k-\omega^2m)^2+c^2\omega^2}}{m\omega^2} \tag{12.11}$$

If Equation 12.10 is rewritten in the following format in terms of Y and Z, then the $m\omega^4YZ$ term in the numerator will have the dimension of power.

$$P = \frac{cm\omega^4YZ}{2\sqrt{(k-\omega^2m)^2+c^2\omega^2}} \tag{12.12}$$

By dividing the power value in Equation 12.12 by the term $m\omega^4YZ$, which has the dimension of power, then dimensionless power [22] can be obtained as

$$P_\text{d} = \frac{P}{m\omega^3YZ_\text{max}} = \frac{\zeta\cdot\eta}{\sqrt{(1-\eta^2)^2+(2\zeta\eta)^2}} \tag{12.13}$$

where ζ denotes the damping ratio, and η is the frequency ratio, and $\frac{c}{m} = 2\zeta\omega_n$, and $\eta = \frac{\omega}{\omega_n}$. The dimensionless power P_d is plotted versus the damping ratio ζ and the frequency ratio η in Figure 12.2.

The dimensionless power, P_d, in Figure 12.2 represents the general characteristics of vibratory systems in which the maximum response occurs at frequency ratio $\eta = 1$ or when frequency of excitation is equal to the natural frequency of the system. Figure 12.2 also illustrates the effect of damping in the harvested power, in which the larger values of the damping ratio correspond to higher magnitudes of power. In an energy harvester, the damping can be replaced by a regenerative energy harvester system. In this case, the damping value is represented by an equivalent damping property of the harvester. The equivalent damping property of energy harvesters is discussed in Section 12.3. It is shown that this equivalent damping can be adjusted to an optimum value in order to optimize the harvested energy while maintaining the general design requirements of the system under investigation.

The actual power value is investigated further using Equation 12.13 as below:

$$P = \frac{m\omega^3YZ_\text{max}\zeta\cdot\eta}{\sqrt{(1-\eta^2)^2+(2\zeta\eta)^2}} \tag{12.14}$$

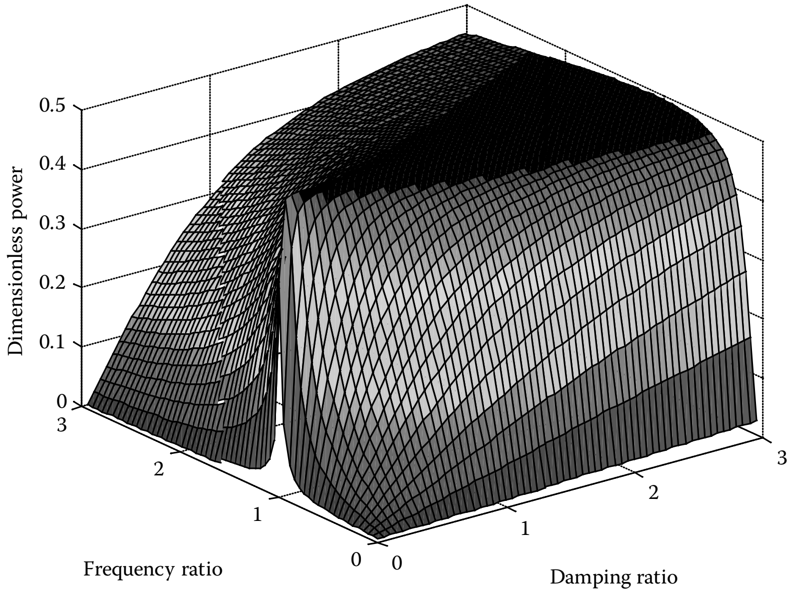


FIGURE 12.2
Dimensionless power versus frequency ratio and damping ratio.

The power P in Equation 12.14 is plotted versus the damping ratio ζ and the frequency f in Figure 12.3, in which $f = \frac{\omega}{2\pi}$. $\omega_n = \sqrt{k/m} = 7.56$ rad/s with k being 16 kN/m and the mass m being 280 kg, as an example. It is observed that the higher power is harvested for larger values of frequency f, where the power is proportional to the cube of the frequency. The power also increases when the damping ratio ζ gets larger. The two important outcomes from the analysis of power include (a) exciting the suspension system with a frequency equal to its natural frequency generates higher power values (Figure 12.2), and (b) the power is proportional to the cubic of the excitation frequency (Figure 12.3).

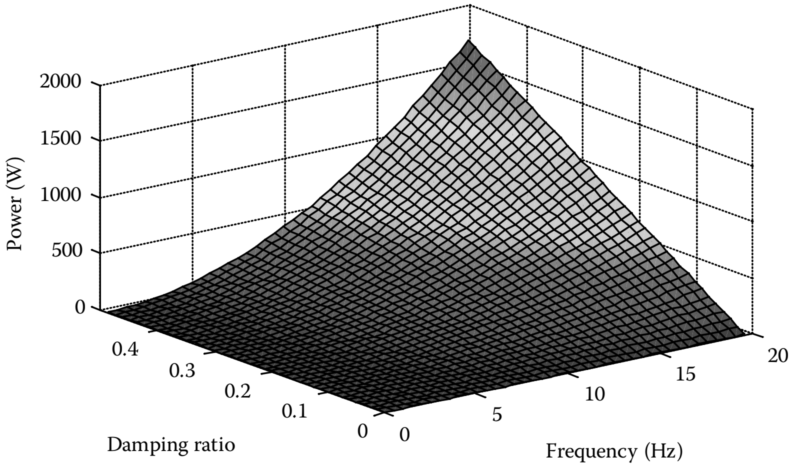


FIGURE 12.3
Harvested power versus frequency and damping ratio.

12.2.1 Energy Conversion Mechanisms

Depending on the application of energy harvesting, the mechanism that most suits the system is selected usually from electrostatic, piezoelectric, and electromagnetic energy conversion mechanisms. In the electrostatic case, electric charges on two boundaries opposing each other can move in a circuit when an external force is applied to the boundaries and therefore convert the kinetic energy of the motion of the boundaries to electrical energy. Such a system can be represented as a capacitance system with two parallel plates as the boundaries. The parallel plates hold the electric charges, and this electric charge can be manipulated when the distance of the plates relative to each other changes. Therefore, the kinetic energy required for moving the plates is proportional to the electrostatic energy of the plates. A schematic of such electrostatic energy conversion is depicted in Figure 12.4.

An ideal capacitor can be modeled by a constant capacitance, C, charge, Q, on each conductor and the voltage, V, between them as $Q = CV$. The amount of energy stored, E, in a capacitor can be calculated as

$$E = \int_0^Q V\,\mathrm{d}q = \frac{1}{2}CV^2 = \frac{1}{2}\frac{Q^2}{C}$$

The capacitance force, F, can be obtained by differentiating the energy with respect to parallel plates variable distance, x, as follows

$$F = \frac{\mathrm{d}E}{\mathrm{d}x} = \frac{1}{2}\frac{\mathrm{d}C}{\mathrm{d}x}V^2 = \frac{1}{2}\frac{\mathrm{d}(Q^2/C)}{\mathrm{d}x}$$

if the capacitance of a parallel plate capacitor can be approximated by $C = \varepsilon_r\varepsilon_0\dfrac{A}{x}$ where A denotes the overlap area of the two plates, ε_r is the dielectric constant of the material between the plates, and ε_0 is the electric constant (8.854×10^{-12} farads per meter). The force can be obtained as $F = \dfrac{1}{2}\dfrac{Q^2}{\varepsilon_r\varepsilon_0 A}$.

In a piezoelectric energy conversion mechanism, when the piezoelectric material is subjected to external force, it can generate voltage in an electrical circuit. Figure 12.5 illustrates a piezoelectric material subjected to external force, F.

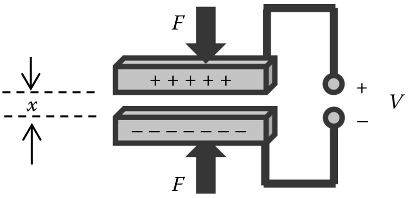


FIGURE 12.4
Electrostatic energy conversion.

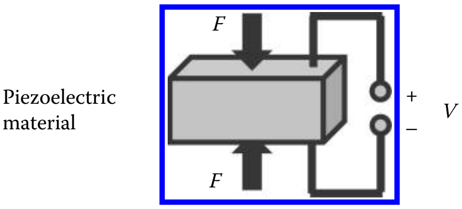


FIGURE 12.5
Piezoelectric energy conversion.

The electromechanical relationship for a linear piezoelectric continuum can be written by employing the principle of conservation of energy using the following equation:

$$\dot{U} = T_{ij}\dot{S}_{ij} + E_i\dot{D}_i$$

Where $\dot{U}$ denotes the rate of change of the stored energy density of the piezoelectric continuum with respect to time, T_{ij} is the stress tensor, $\dot{S}_{ij}$ is the rate of the change of the strain tensor with respect to time, E_i is the electrical field tensor, $\dot{D}_i$ represents the rate of change of the electric displacement tensor with respect to time.

If the electric enthalpy density H is expressed as $H = U - E_iD_i$. Then the derivative of electric enthalpy density with respect to time can be obtained by

$$\dot{H} = T_{ij}\dot{S}_{ij} - D_i\dot{E}_i,$$

which shows that electric enthalpy density is a function of strain and electrical field, and therefore,

$$T_{ij} = \frac{\partial H}{\partial S_{ij}} \tag{12.15}$$

$$D_i = -\frac{\partial H}{\partial E_i} \tag{12.16}$$

In the linearized theory of piezoelectricity, the electric enthalpy can be given by

$$H = \frac{1}{2}c^E_{ijkl}S_{ij}S_{kl} - e_{kij}E_kS_{ij} - \frac{1}{2}\varepsilon^S_{ij}E_iE_j \tag{12.17}$$

where c^E_{ijkl} denotes the elastic constant at constant electric field, E (the superscripts), ε^S_{ij} is the permittivity constant for a constant strain, S (the superscripts), and e_{kij} is the piezoelectric constant.

Substituting Equation 12.17 into Equation 12.15 gives

$$T_{ij} = c^E_{ijkl}S_{kl} - e_{kij}E_k \tag{12.18}$$

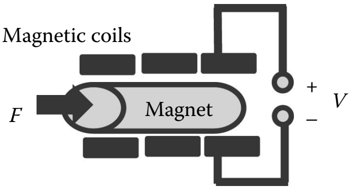


FIGURE 12.6
Electromagnetic energy conversion.

and using Equation 12.17 into Equation 12.16 gives

$$D_i = e_{ikl}S_{kl} + \varepsilon_{ik}^{S}E_k \tag{12.19}$$

where Kronecker delta δ_{ij} is employed for the relationship $\partial S_{ij}/\partial S_{ji}$, which gives 1 for $i = j$ and gives zero otherwise. Equations 12.18 and 12.19 represent the linear constitutive equations for the piezoelectric continuum.

In an electromagnetic energy conversion mechanism, the kinetic energy of a moving magnet inside a magnetic coil can be converted to electrical energy (see Figure 12.6). An electric motor that acts as an electromagnetic energy convertor can convert mechanical to electrical energy and vice versa. A linear actuator can be built based on the concept of electric motors in which the motion of the moving magnet is linear rather than the conventional rotational motion in rotary motors. The equation of a linear actuator is considered here as it is appropriate for the applications discussed in the following sections.

If voltage, V, is applied to a linear actuator, it gives the moving magnet the velocity of, $\dot{z}$, as given in the following equation [7]:

$$V = -k_v\dot{z} \tag{12.20}$$

where k_v is called the motor voltage constant.

If i is the electric current in the armature, and r is the resistance of the armature applied to the magnetic coil in the actuator, it can provide the actuation force, F, given by the following expression:

$$F = k_t i \tag{12.21}$$

where k_t is called the motor torque constant.

Substituting Equation 12.20 into Equation 12.21 gives the relationship between force and velocity as below:

$$F = -\frac{k_v k_t}{r}\dot{z} \tag{12.22}$$

12.3 Self-Powered Dynamic Systems

A self-powered dynamic system [1] is defined as a dynamic system powered by its own excessive kinetic energy, renewable energy, or a combination of both. The particular area of work is the concept of fully or partially self-powered dynamic systems requiring zero or reduced external energy inputs. The technologies exploited in this chapter are particularly associated with self-powered sensors, regenerative actuators, human powered devices, and a solar-powered dynamic system as self-sustained systems. Various strategies can be employed to improve the design of a self-powered system, and among them, adopting a bio-inspired design is investigated to demonstrate the advantage of biomimetics in improving power density.

12.3.1 Concept of Self-Powered Dynamic Systems

The concept of self-powered dynamic systems is illustrated in Figure 12.7, which is explained below [1]:

1. Input power or input excitation force to the system (e.g., fuel energy or renewable energy powering a vehicle system, or wind/earthquake forces to a structure).
2. The kinetic energy in the direction of motion of a dynamic system is only recovered when the system is stationery (e.g., a bridge structure) or when the recovery of the energy does not affect the performance of the system (i.e., is negligible in comparison with the power required for creating motion, e.g., a low-powered sensor).

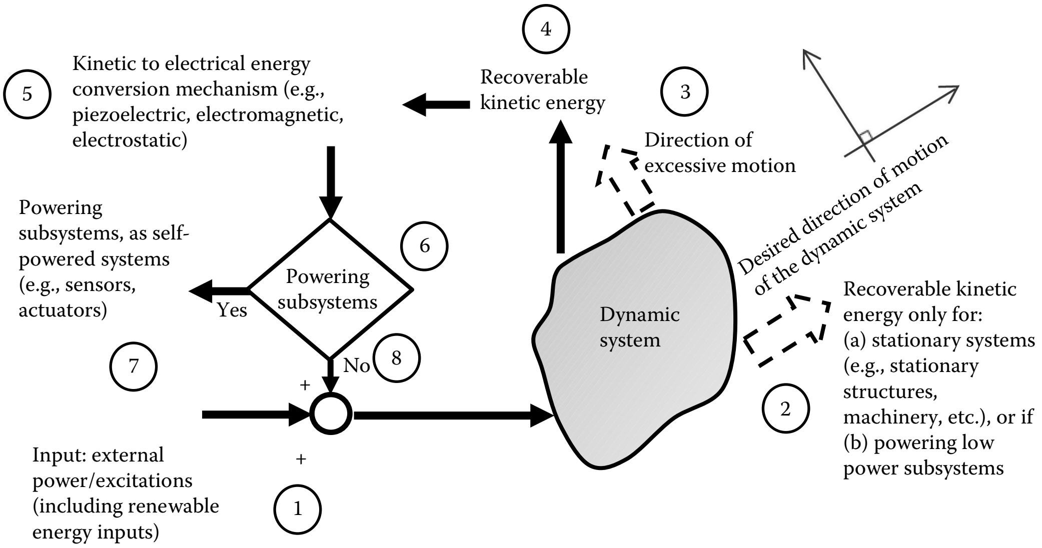

④
Recoverable kinetic energy
⑤
Kinetic to electrical energy conversion mechanism (e.g., piezoelectric, electromagnetic, electrostatic)
③
Direction of excessive motion
Desired direction of motion of the dynamic system
Powering subsystems, as self-powered systems (e.g., sensors, actuators)
⑥
Powering subsystems
Yes
No
⑧
⑦
+
+
Dynamic system
Recoverable kinetic energy only for:
(a) stationary systems (e.g., stationary structures, machinery, etc.), or if
(b) powering low power subsystems
②
Input: external power/excitations (including renewable energy inputs)
①

FIGURE 12.7
The concept of self-powered dynamic systems. (From Khoshnoud, F., Y. K. Chen, R. K. Calay, C. W. de Silva, and H. Owhadi, "Self-powered dynamic systems," *European Conference for Aeronautics and Space Sciences*, Munich, Germany, Paper No. 275, July 1–5, 2013.)

3. The motion of the dynamic system perpendicular to the desired direction of the motion is usually the wasted kinetic energy in the system (e.g., the vertical motion of a road vehicle suspension or vibration of an aircraft wing).
4. The vertical motion of the dynamic system is a source of recoverable kinetic energy.
5. The recoverable kinetic energy can be converted to electrical energy through an energy conversion mechanism, such as electromagnetic, piezoelectric, or electrostatic schemes. Examples include replacing the viscous damper of a car shock absorber with a regenerative actuator, embedding piezoelectric material in aircraft wings, or the vibration of a micro cantilever in a MEMS sensor.
6. The recovered electrical power can be stored or used directly as a power source.
7. The recovered electrical energy can power subsystems of the dynamic system, such as sensors and actuators.
8. The recovered electrical power can be realized as an input to the dynamic system itself.

The succeeding section provides the theoretical background associated with the concept of self-powered dynamic systems.

12.3.2 Theory of Self-Powered Systems

A self-powered system consisting of a regenerative scheme is illustrated in Figure 12.8. This is an example of a regenerative system with an electromagnetic energy conversion mechanism. The experimental rig in the figure consists of mass, a spring, and an

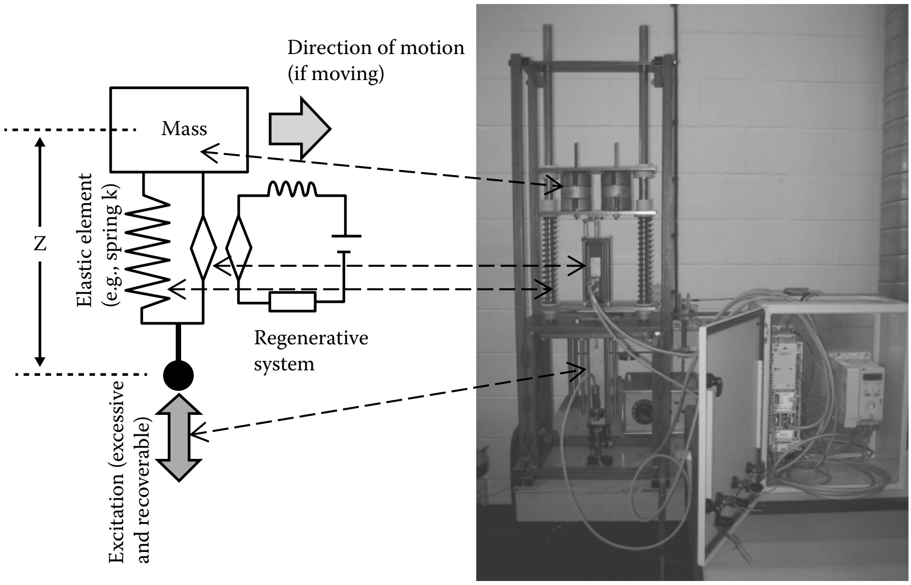


FIGURE 12.8
Representation of a self-powered dynamic system experimental rig. (From Khoshnoud, F., Y. K. Chen, R. K. Calay, C. W. de Silva, and H. Owhadi, "Self-powered dynamic systems," *European Conference for Aeronautics and Space Sciences*, Munich, Germany, Paper No. 275, July 1–5, 2013.)

electromagnetic regenerative actuator. The input excitation is generated using a shaker as in the figure. A controller system is designed to adjust the regenerative system for various modes of operation. The operation modes of a regenerative system can include drive, regeneration, and brake modes. The drive mode is developed to control the motion, which can use the recovered power from the system for its actuation. The brake mode damps the vibration, which can be considered as an equivalent damping for the system provided by the electrical properties of the actuator control system. In the regeneration mode, the kinetic energy is converted to electrical energy. A practical example for application of this rig can be referred to as a self-powered regenerative shock absorber in vehicle or a self-powered sensor (e.g., [1] and [34]).

Equations 12.12 and 12.13 represent the analytical formulation of power associated with the amount of energy that can be harvested from kinetic energy. Figures 12.2 and 12.3 represent the dimensionless power and the harvested power, respectively.

As discussed in the electromagnetic energy conversion scheme in Section 12.2.1, for a DC linear motor/actuator, the relationship between force and velocity in terms of the motor parameters can be determined as [7] and [34] follows. From Equations 12.20 and 12.21, the voltage, V, and force, F, can be expressed by the following equations:

$$V = -k_\mathrm{a}\dot{z} \tag{12.23}$$

$$F = k_\mathrm{a} i \tag{12.24}$$

Note that in a DC motor the following assumption can be considered:

$$k_v = k_t = k_\mathrm{a}$$

From Equations 12.23 and 12.24, the force, F, that an actuator generates can be obtained as a function of velocity in terms of motor constant, k_a, resistance, r, and velocity $\dot{z}$.

$$F = -\frac{k_\mathrm{a}^2}{r}\dot{z} \tag{12.25}$$

If the voltage of the power source is V_p, then the force in the actuator can be obtained as

$$F = k_\mathrm{a}\frac{V_\mathrm{p} - k_\mathrm{a}\dot{z}}{r} \tag{12.26}$$

The amount of power consumed by the voltage source in order to generate the force, F, by the actuator, can be given by

$$P_c = V_\mathrm{p} i = \left(\frac{rF + k_\mathrm{a}\dot{z}}{k_\mathrm{a}}\right)\frac{F}{k_\mathrm{a}} \tag{12.27}$$

By comparing Equation 12.27 with the force in a viscous damper, the equivalent damping for the motor can be stated as

$$c_{\text{eq}} = -\frac{k_{\text{a}}^2}{r}$$

And, therefore, the consumed power can be written in terms of the equivalent damping as

$$P_c = V_{\text{p}} i = \frac{1}{c_{\text{eq}}} F^2 + F\dot{z} \tag{12.28}$$

If λ is defined by the following expression (for $\dot{z} \neq 0$),

$$\lambda = \frac{F}{-c_{\text{eq}}\dot{z}}$$

then, the power in Equation 12.28 can be rewritten as

$$P_c = c_{\text{eq}}\dot{z}^2\lambda(\lambda - 1) \tag{12.29}$$

Figure 12.9 shows the normalized power, $P_c/c_{\text{eq}}\dot{z}^2$, versus λ, which represent the power consumption and power generation scenarios of a regenerative system. The regeneration of energy corresponds to the negative region of the plot (where $P_c/c_{\text{eq}}\dot{z}^2 < 0$), and the power consumption corresponds to the positive values of the curve in Figure 12.9.

The surface in Figure 12.10 shows P_c in Equation 12.29 versus λ and $\dot{z}$. In the calculation of power in this figure, it is assumed that $c_{\text{eq}} = 2\zeta\omega_n m$, for $\zeta = 0.3$ and $\omega_n = (k/m)^{1/2}$, which corresponds to parameters selected for dynamic characteristics of a regenerative system

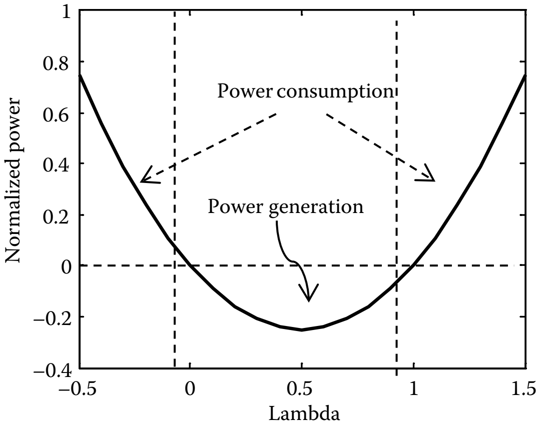


FIGURE 12.9
Normalized power showing power consumption and power generation by a regenerative system.

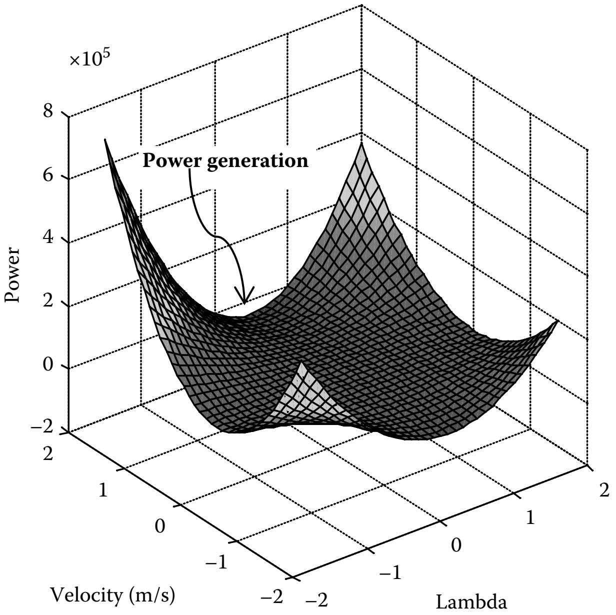


FIGURE 12.10
Power versus lambda and velocity.

for a quarter of a vehicle model as an example [1]. The region of the surface in Figure 12.10 in which $P_c < 0$ (negative power values) corresponds to the power generation state, and the positive values correspond to power consumption by the system.

For $|F| < |c_{eq}\dot{z}|$, or $0 < \lambda < 1$ region (Figures 12.9 and 12.10), the required force to drive the actuator is less than the dynamic force $(c_{eq}\dot{z})$ in the actuator. The dynamic force, $c_{eq}\dot{z}$, is due to the kinetic energy of the actuator motion generated by external excitation forces applied to the system (e.g., vibration of a vehicle during driving, which is an excessive kinetic energy in the system). Therefore, the kinetic energy can be converted to electrical energy. This electrical energy can be fed back to the actuator system, which generates the actuation driving force to control the vibration as a self-powered mechanism (Figure 12.7).

The theory addressed below is applicable to self-powered systems, including sensors, actuators (e.g., [10,34]), etc. The energy conversion mechanism can be chosen as electromagnetic, piezoelectric, electrostatic, etc., based on what is most suitable to the application under investigation, and subsequently, the electromechanical properties of the energy conversion mechanisms are chosen accordingly.

12.3.3 Renewable Energy for Dynamic Systems

As discussed above, a source of power for a self-powered dynamic system is considered as renewable energy. A solar-powered airship is considered here as an example of a self-sustained dynamic system, which is fully powered only by renewable energy sources. The Multibody Advanced Airship for Transport (MAAT) [68–72] (Figure 12.11) is a self-sustained system powered by solar energy. An airship, unlike aircraft, requires a large volume of lighter-than-air gas in order to generate enough lift by buoyancy for the airship. Requirement of this large volume provides a large surface area, which can be ideal for

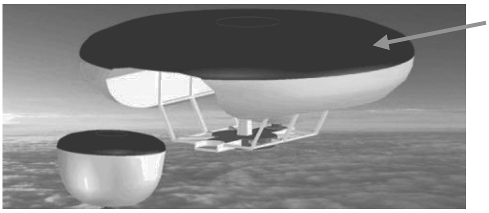


FIGURE 12.11
A solar-powered airship based on the concept of MAAT (http://www.eumaat.info).

installing a large area of photovoltaic (PV) solar panels on the airship roof. This system uses the solar power and converts it to electrical energy for daytime operations and also generates hydrogen using electrical energy to be used by a fuel cell system as a power supply during the night. The large area of PV cells on the airship can provide sufficient energy for the entire operation of the airship flight and other energy demands for the airship systems. In particular MAAT is based on the concept of multibody airships, which allow more flexibility for transport compared with existing airships. The smaller airships (called feeders) can travel individually from the main airship (called cruisers), which allows multiple operations and missions with a multibody airship system (Figure 12.11).

The dynamic behavior of the airship system is given below [68]. The buoyancy force in an airship is equivalent to the weight of displaced air given by

$$B = U_{\mathrm{H}}\rho(h)g \tag{12.30}$$

where U_{H} is hull volume, $\rho(h)$ is atmospheric density at altitude h, and g is gravity.

The required thrust by the airship can be expressed by

$$T_{\mathrm{req}} = D = \frac{1}{2}\rho(h)V^2U_{\mathrm{H}}^{2/3}C_D \tag{12.31}$$

where D denotes aerodynamic drag force, V is the airship wind-relative velocity, and C_D is the drag coefficient. The required power for this thrust is given by

$$P_{\mathrm{req}} = \frac{T_{\mathrm{req}}}{\eta} \tag{12.32}$$

where η denotes combined efficiency of the propeller and the motor.

A free-body diagram of the forces in the x–z plane, including aerodynamic lift L, drag D, thrust T, weight W, and buoyancy B are illustrated in Figure 12.12.

From Figure 12.12, the sum of the forces applied to the airship in the wind-relative frame can be expressed as

$$\begin{aligned}\Sigma \boldsymbol{F} &= \left[(B-W)\sin\gamma + T\cos(\alpha+\mu) - D\right]\vec{x}\\ &\quad + S\vec{y} + [(W-B)\cos\gamma - L - T\sin(\alpha+\mu)]\vec{z}\end{aligned} \tag{12.33}$$

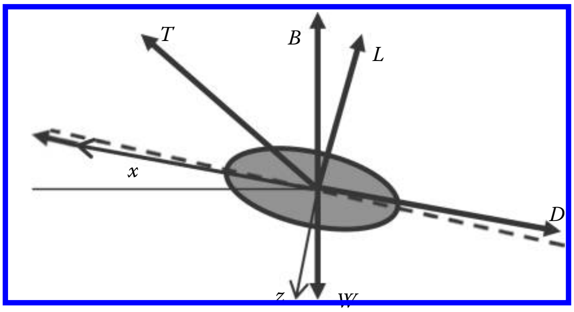


FIGURE 12.12
External forces acting on the airship.

where γ is the angle between the x axis and the horizontal line, α is the angle of attack, which is the angle between the orientation of the airship (dashed line in Figure 12.12) and the x axis, and μ is the angle between the thrust T and the dashed line. The lateral force S is perpendicular to the plane of the paper.

The lift force can be expressed by $L = qU_{\mathrm{H}}^{2/3}C_L(\alpha)$ and the drag force is given by $D = qU_{\mathrm{H}}^{2/3}C_D(\alpha)$. Here, $C_L(\alpha)$ and $C_D(\alpha)$ are the lift and drag coefficients, respectively, which are functions of the angle of attack.

12.3.3.1 Power System

Solar energy is the main source of power for the airship propulsion system, avionics, control, and payload (Figure 12.13). Solar power is also used to generate hydrogen for a fuel cell system to be used during the night when solar energy is not available. When electricity from the PVs is supplied to the fuel cell system, the fuel cell generates hydrogen, and conversely, when hydrogen is supplied to the fuel cell, it generates electrical energy for the airship system. Figure 12.13 illustrates the power and control system of the self-sustained airship.

In an airship system, there are various parameters that are controlled to ensure complete operation and missions of the airship. Flight control is one of the control problems, which include hovering, maneuvering, and trajectory path tracking operations. An integrated power, control, and propulsion system of the airship in Figure 12.13 illustrates a trajectory control application as an example of the control system objective. In this figure, an overall integrated power, control, and propulsion system is depicted in which the position, orientation, and velocity sensors are used to control the trajectory of the airship dynamics through a feedback control system. The solar energy absorbed by the solar panel is fed to a controlled switch box, which decides to either provide solar power to the propulsion system, control system, actuators, avionics, and payload directly or supply power to the fuel cell for generating and storing hydrogen. Therefore, there is a two-way information and energy exchange between the power and the control system, which acts as a self-sustained system. The airship dynamic model predicts the behavior of the airship. The behavior of the airship (e.g., position and velocity) is sensed by velocity, position, and orientation sensors. Based on the dynamic model of the airship and the error between the desired output and measured output (by sensors), the control system calculates the amount of power input.

The power generated by the photovoltaic (PV) cells depends on the value and spectral distribution of irradiance in the plane of solar cell and the temperature. Irradiance

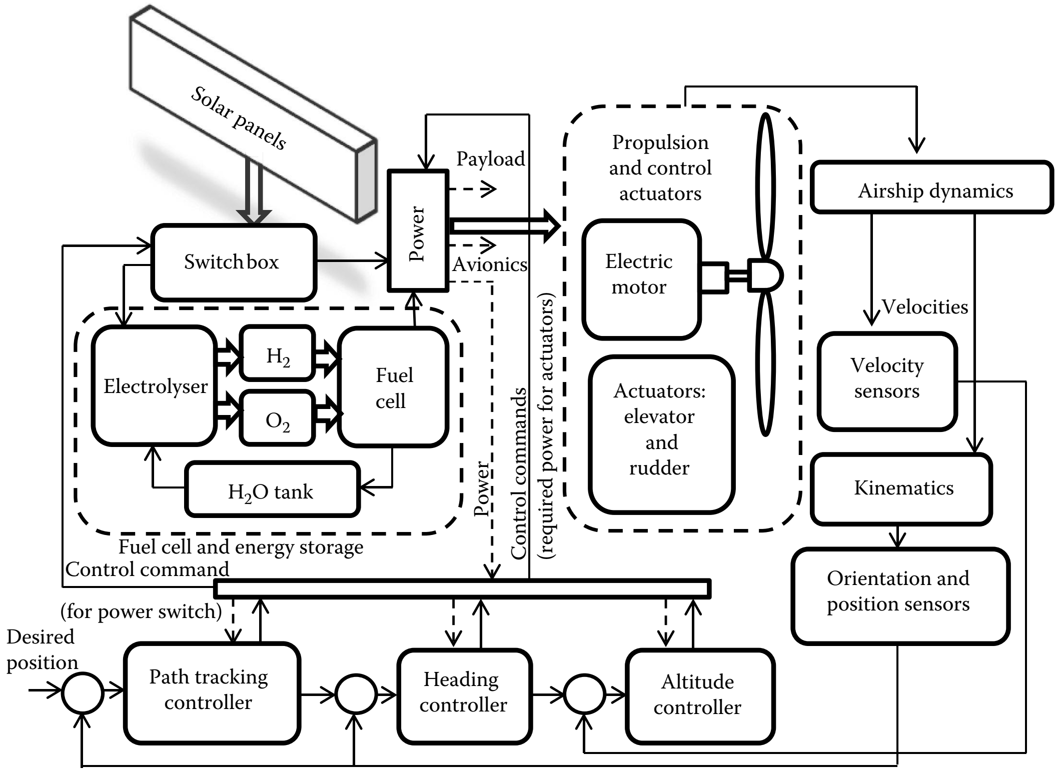

Solar panels
Payload
Power
Avionics
Switch box
Propulsion
and control
actuators
Electric
motor
Airship dynamics
Velocities
Velocity
sensors
Electrolyser
H_2
O_2
Fuel
cell
H_2O tank
Actuators:
elevator
and
rudder
Kinematics
Power
Control commands
(required power for actuators)
Fuel cell and energy storage
Control command
(for power switch)
Orientation and
position sensors
Desired
position
Path tracking
controller
Heading
controller
Altitude
controller

FIGURE 12.13
A solar fuel cell–powered airship system.

absolute value is a function of altitude, which increases with height. When the Earth–Sun distance changes at different times during the year, the irradiance value varies between 1320 and 1415 W/m^2. The incident solar radiation is related to latitude, time of year, airship orientation, and flight altitude, wind speed, and ambient temperature, which all influence the convective heat transfer of the arrays.

The electrical power system of a PV device can be represented by the equivalent circuit in Figure 12.14 [73–75].

The current–voltage relationship (the I–V curve) of the PV device in Figure 12.14 at a fixed cell temperature and solar radiation can be obtained in terms of the diode current, I_D, the light current, I_L, the diode reverse saturation current, I_o, the series resistance, R_s, and shunt resistance, R_{sh} as given in the following equation:

$$I = I_L - I_o\left[e^{\frac{V+IR_s}{a}} - 1\right] - \frac{V + IR_s}{R_{sh}}$$

where the factor, a, called the ideality factor is $a \equiv N_s n_I kT_c/q$, and q is the electron charge, k denotes the Boltzmann's constant, n_I is the usual ideality factor, N_s is the number of cells in series, and T_c is the cell temperature. The amount of power, P, can be calculated as $P = I \times V$. The direct normal radiation at the range of 8 to 15 km height can be calculated by [76]

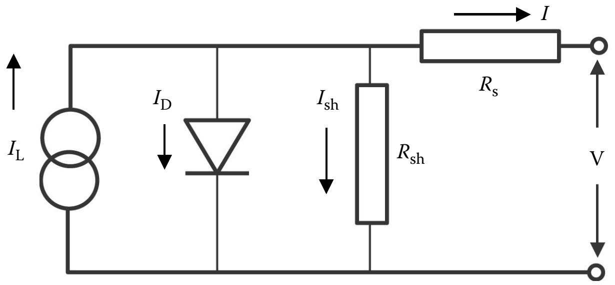


FIGURE 12.14
Equivalent circuit of a photovoltaic device.

$I_n(A) = 1.353e^{-0.357\left(\frac{1}{\sin A}\right)^{0.678}}$ where A is the solar altitude and A = secλ, where λ denotes the zenith distance. The efficiency of solar cells are normally lower than 40% and, in most cases, in the range of 10%–20% for commercially available solar cells including crystalline silicon cells (Si), amorphous silicon cells (a-Si), gallium arsenide cells (GaAs), and polycrystalline cells, such as cadmium sulphide (CdS). The efficiency of the solar power system depends on various factors, including solar cell packing area, conversion efficiency, electrical component efficiency, and propulsive efficiency.

12.3.4 Human-Powered Systems

Another source of renewable energy is the kinetic energy associated with human motion. Human motion during everyday life can be converted to electrical energy using energy harvesting conversion mechanisms. For instance, kinetic energy due to walking using a shoe insole can generate up to 220 V of voltage and 40 μA of current [77]. The force, F, applied to the ground when walking can be represented by $F = \alpha \times Mg$ where the factor α depends on the speed of walking, M is the mass of the person, and g is the gravity. α is equal to 1 for a quasistatic loading when the inertial forces are negligible and when the person is standing. When walking, α value can be in the range of 1 to 1.5 [78], depending on how fast the person is walking, and when running and jumping, the α value can reach up to 2 or higher. If the person is walking on a strip of piezoelectric material and exerts the amount of 1 mm compression displacement, d, on the piezoelectric material, then the amount of energy that can generate in each step of walking can be calculated as $E_{\text{step}} = \eta \times \alpha \times Mg \times d$ where η is the efficiency of the piezoelectric energy harvesting mechanism. If the distance between each step is considered as 0.5 m, then walking a distance of L, for n number of people and for h hours of time can give the total amount of energy in kilowatt hours as $E_{\text{total}} = n \times E_{\text{step}} \times \frac{L}{0.5} \times \frac{1\text{ kWh}}{3.6 \times 10^6\text{ J}}$.

The application of a system that generates electricity from the kinetic energy of walking or human motion can be considered as a power source to self-powered systems, such as for powering a lighting system along a walking path or powering wireless sensors for detecting motion. An example would be an intelligent floor that can generate power to a wireless sensor when a patient or an elderly person falls on the floor at home or in a hospital. Such a system can send an alert signal to the caregivers.

The instrumented bicycle in Figure 12.15 can show the amount of power that can be generated by a person's pedaling activity. The experimental investigation is carried out

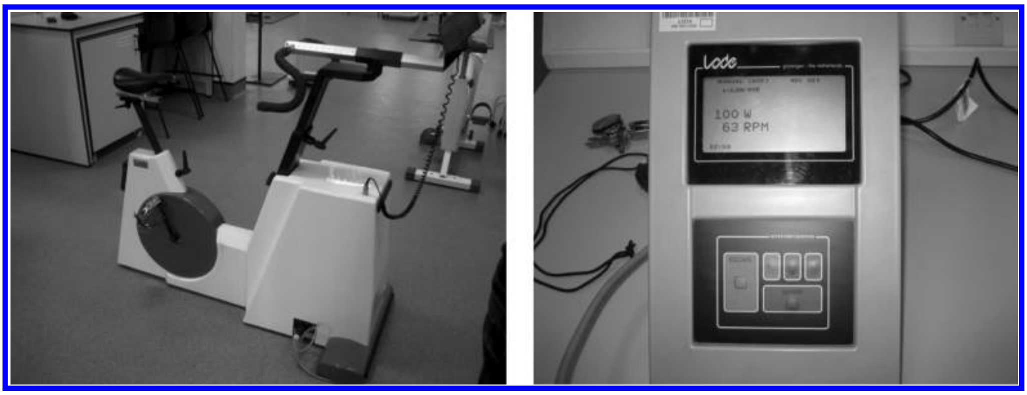


FIGURE 12.15
An instrumented bike facility for measuring the amount of power generated by human motion.

by setting the power output to a fixed value (e.g., 50 W, 100 W, 200 W, etc.) by the system. Therefore, the generated power produced by the bicycle is fixed to a certain value regardless of the person's input power and speed of input to the system as long as the system is running. The bicycle system makes the activity harder when at low speed pedaling and easier at high speed pedaling and by doing that adjusts the power output to a fixed value [1].

The duration of the generated power depends on the heart rate. The formula to obtain this duration time is

The theoretical maximum heart rate of a person =
220 minus the age of the cyclist in years

If the heart rate of a person is equal to 75% to 85% of the theoretical maximum heart rate, then the person can continue the activity for about 10 to 15 min. For a heart rate of 60% to 75% of the maximum calculated heart rate, an activity of about 30 min to 1 hr is predicted. For a heart rate of less than 60% of the theoretical maximum, the person can keep up the activity with this heart rate for a long time unless the individual is fatigued.

12.4 Bio-Inspired Dynamic Systems

Inspiration from nature allows the design of more efficient engineering systems. Biological systems and animals are products of nature, which have evolved to better respond to life activities. For instance, studies carried out on the mechanics of aquatic animals by biologists have shown that even a dead trout can swim upstream by extracting energy and generating thrust-producing body deflections, and that whale flukes are capable of absorbing energy from surface waves and creating thrust whether the whale is alive or dead [55–58]. A fish is capable of utilizing energy in upstream vortices to enhance its locomotion performance and efficiency. It was demonstrated that "through passive body deformations and vorticity control, a fish could even propel itself forward inside a Kármán vortex street without energy expense" [59]. This is the motivation for employing biomimetics to improve engineering systems design. For instance, a piezoelectric energy harvester in a bio-inspired array of harvesters (when placing piezoelectric cantilevers next to each other in configuration analogues to a group of fish swimming next to each other) can generate more power compared to a piezoelectric energy harvester in isolation [60–63]. Vertical axis wind turbines designed on the basis of fish schooling shows an order of magnitude improvement in the power density of the wind farms [64]. Accordingly, improving the energy efficiency of a piezoelectric energy harvesting system is discussed below. Figure 12.16 illustrates the analogy between fish schooling and piezoelectric energy harvesters. This piezoelectric energy harvesting scheme is designed on the basis of aquatic animal locomotion. The piezoelectric material embedded in the cantilevers, as shown in Figure 12.16, converts the kinetic energy due to the vibration of the cantilevers to electrical energy.

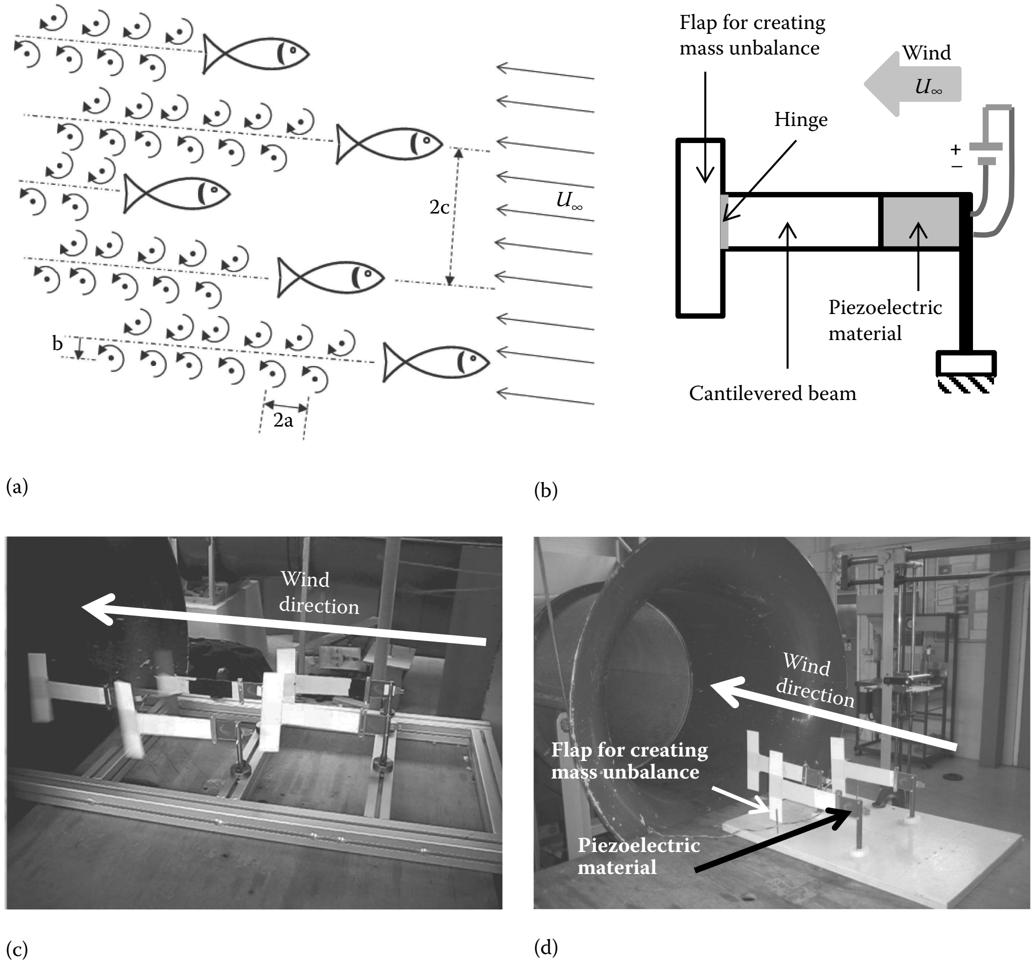
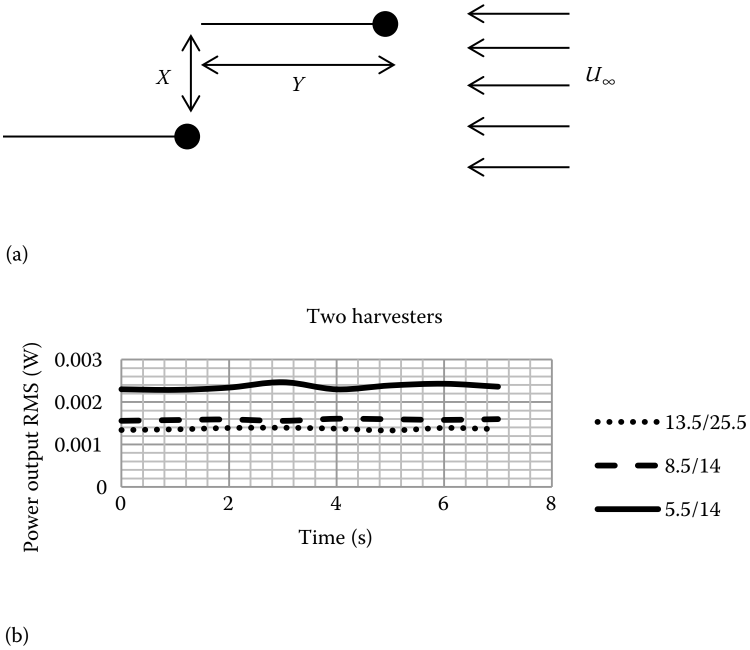

2c
U_∞
b
2a
(a)
Flap for creating mass unbalance
Wind
U_∞
Hinge
+
−
Piezoelectric material
Cantilevered beam
(b)
Wind direction
(c)
Wind direction
Flap for creating mass unbalance
Piezoelectric material
(d)

FIGURE 12.16
Bio-inspired piezoelectric energy harvesters: (a) fish schooling and the generated vortices from fish locomotion, (b) a piezoelectric cantilever, (c) bio-inspired piezoelectric energy harvesters—experimental setup in the wind tunnel using five piezoelectric cantilevers, which corresponds to the five fish in the fish schooling, and (d) piezoelectric energy harvesters in the wind tunnel with another arrangement of three piezoelectric cantilevers.



(a)

(b)

FIGURE 12.17
(a) The top view of two piezoelectric cantilevers and (b) experimental power output for the bioinspired piezoelectric energy harvesters.

The flap that is hinged to the cantilevered beam creates mass unbalance when the cantilever flutters. The power density of this array of piezoelectric energy harvesters can be improved by arranging them at a certain distance relative to each other as inspired by how fish swim in groups to use less muscle energy. The harvested energy from these energy harvesters can be utilized to power electronic components, such as sensors. The energy supply is a renewable energy source (i.e., wind energy), and therefore, the sensor will not require any battery or external power source. Hence, the device operates as a self-powered system based on renewable energy.

A sample experimental result of such bio-inspired energy harvesters is given in Figure 12.17 [1]. The streamwise distance of $Y = 14$ cm and 25.5 cm and cross-stream distance of $X = 5.5$ cm, 8.5 cm, and 13.5 cm are chosen in this experiment. The air speed in this example is $U_\infty = 18$ m/s. It is obtained that there is an optimum separation between the cantilevers where an energy harvester can generate the maximum power (Figure 12.17). This is due to the interaction of harvesters with each other through the flutter-induced vortices in the fluid flow. Therefore, a cantilever device harvests the dynamic energy in the fluid flow generated by other energy harvesters, analogous to how fish swim in groups or birds fly in formation to use less muscle energy.

12.4.1 Piezoelecteric Energy Harvesting from Aeroelastic Vibrations

This section provides the theoretical background of aeroelastic vibration for piezoelectric energy harvesting of a two-degree-of-freedom lumped parameter model when exposed to uniform incompressible potential flow. The two degrees of freedom are the plunge displacement or translation h and the pitch displacement or rotation α in Figure 12.18, which

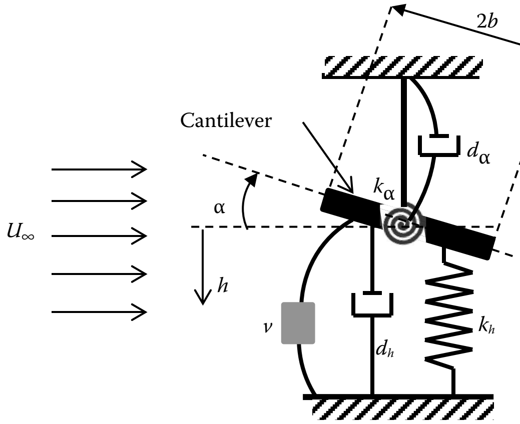


FIGURE 12.18
A piezoaeroelastic section exposed to uniform airflow.

are defined using aircraft wing flutter analysis. The plunge degree of freedom is responsible for the input excitation of the piezoelectric material, which is connected to the cantilevered beam.

Figure 12.18 shows the piezoaeroelastic section exposed to a uniform incompressible airflow with the speed of U_∞. In this figure, v represents the voltage output of the piezoelectric material across an external electric load. The airfoil angle of attack is denoted by α. The cantilever starts to flutter when U_∞ reaches the aeroelastic flutter instability of the cantilever. The equation of motion of the cantilever in Figure 12.18 can be written using Hamilton's principle as

$$\int_{t_2}^{t_1} (\delta T - \delta U + \delta W_{\text{ie}} + \delta W_{\text{nce}} + \delta W_{\text{nca}} + \delta W_{\text{sd}})\,\text{d}t = 0 \tag{12.34}$$

where T and U are the total kinetic and potential energies per length. W_{ie} denotes the internal electrical energy per length, W_{nce} is the virtual work component of the nonconservative electrical charge, W_{nca} is the virtual work component of the aerodynamic loads, and W_{sd} is the virtual work of the structural damping.

The kinetic energy of the system per length can be written as

$$T = \frac{1}{2}m(\dot{h}^2 + 2x_\alpha b\dot{h}\dot{\alpha}) + \frac{1}{2}m_{\text{f}}\dot{h}^2 + \frac{1}{2}I_{\text{p}}\dot{\alpha}^2 \tag{12.35}$$

where m is the airfoil mass per length, m_{f} is the fixture mass per length connecting the airfoil to the plunge springs, x_α is the chordwise offset from the reference point to the centroid, b is the semichord length, and I_{p} is the moment of inertia per length about the reference point. The potential energy of the system can be given by

$$U = \frac{1}{2}k_h h^2 + \frac{1}{2}k_\alpha \alpha^2 - \frac{1}{2}\frac{\theta}{l}hv \tag{12.36}$$

where k_h is the stiffness per length associated with the plunge degree of freedom, k_α is the stiffness in pitch degree of freedom, θ is the electromechanical coupling term, l is the span length of the cantilever, and v is voltage. The internal electrical energy is

$$W_{\text{ie}} = \frac{1}{2}\frac{C_{\text{p}}^{\text{eq}}}{l}v^2 + \frac{1}{2}\frac{\theta}{l}hv \tag{12.37}$$

where C_{p}^{eq} is the equivalent capacitance of the piezoelectric.

The virtual work components can be determined as

$$\delta W_{\text{nce}} = \frac{Q}{l}\delta v \tag{12.38}$$

$$\delta W_{\text{nca}} = -L\delta h + M\delta\alpha \tag{12.39}$$

$$\delta W_{\text{sd}} = -d_h\dot{h}\delta h - d_\alpha\dot{\alpha}\delta\alpha \tag{12.40}$$

where Q is the output of the electrical charge, L is the aerodynamic lift per length, M is the aerodynamic pitching moment per length, d_h is the structural damping coefficient per length for the plunge DOF, and d_α is the structural damping coefficient per length for the pitch DOF.

Harmonic response assumption gives $h = \bar{h}e^{j\omega t}$, $\alpha = \bar{\alpha}e^{j\omega t}$, $v = \bar{v}e^{j\omega t}$, $L = \bar{L}e^{j\omega t}$, $M = \bar{M}e^{j\omega t}$. Using the Lagrange equations with harmonic response assumption and the Equations 12.35 through 12.40 will lead to the complex eigenvalue problem below [12].

$$\begin{bmatrix} \left[\beta + \frac{l_h}{\mu} - \kappa(\omega) - \sigma^2(1 + j\gamma_h)\lambda\right] & \left(x_\alpha + \frac{l_a}{\mu}\right) \\ \left(x_a + \frac{m_h}{\mu}\right) & \left[r^2 + \frac{m_\alpha}{\mu} - r^2(1 + j\gamma_\alpha)\lambda\right] \end{bmatrix} \begin{bmatrix} \frac{\bar{h}}{b} \\ \bar{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{12.41}$$

Here, λ, σ, r, μ, and β are dimensionless terms denoting complex eigenvalue ($\lambda = (\omega_\alpha/\omega)^2$), frequency ratio ($\sigma = \omega_h/\omega_\alpha$), radius of gyration $\left(r = \sqrt{I_{\text{p}}/mb^2}\right)$, ratio of the airfoil to affected air mass ($\mu = m/\pi\rho_\infty b^2$), and mass ration ($\beta = (m + m_f)/m$), where $\omega_h = \sqrt{k_h/m}$, $\omega_\alpha = \sqrt{k_\alpha/I_{\text{P}}}$, ρ_∞ is the free-stream air mass density, $\gamma_h = d_h\omega/k_h$, $\gamma_\alpha = d_\alpha\omega/k_\alpha$, and ℓ_h, ℓ_α, m_h, m_α are the components of the aerodynamic loads, given by

$$\ell_h = 1 - j\frac{2}{k}C(k) \tag{12.42}$$

$$\ell_\alpha = -a - j\frac{1}{k} - \frac{2}{k^2}C(k) - j\frac{2}{k}\left(\frac{1}{2} - a\right)C(k) \tag{12.43}$$

$$m_h = -a + j\frac{2}{k}\left(\frac{1}{2} + a\right)C(k) \tag{12.44}$$

$$m_a = \frac{1}{8} + a^2 - j\frac{1}{k}\left(\frac{1}{2} - a\right) + \frac{2}{k^2}\left(\frac{1}{2} + a\right)C(k) + j\frac{2}{k}\left(\frac{1}{4} - a^2\right)C(k) \tag{12.45}$$

where α is the dimensionless location of the reference point, k is the reduced frequency, $C(k)$ is the Theodorsen function, and $H_n^{(2)}(k)$ are Hankel functions, which are given by

$$k = \frac{b\omega}{U} \tag{12.46}$$

$$C(k) = \frac{H_1^{(2)}(k)}{H_1^{(2)}(k) + jH_0^{(2)}(k)} \tag{12.47}$$

$$H_n^{(2)}(k) = J_n(k) - jY_n(k) \tag{12.48}$$

where $J_n(k)$ is the Bessel function of the first kind, and $Y_n(k)$ are Bessel functions of the second kind.

The term $\kappa(\omega)$ is

$$\kappa(\omega) = \frac{j\theta^2}{\omega m\ell\left(j\omega C_{\mathrm{P}}^{\mathrm{eq}} + \dfrac{1}{R_l}\right)} \tag{12.49}$$

where R_l is the external electrical load resistance, and $\dot{Q} = v/R_l$.

An iterative eigenvalue procedure is employed (e.g., using P-K method) for eigensolution of Equation 12.41, which obtains $\bar{h}$ and $\bar{\alpha}$, and therefore, $\bar{v}$ is determined by

$$\bar{v} = \frac{-j\omega\theta\bar{h}}{j\omega C_{\mathrm{P}}^{\mathrm{eq}} + \dfrac{1}{R_l}} \tag{12.50}$$

The amount of electrical power can be calculated using $\bar{v}^2/R_l$.

When the piezoelecteric energy harvesters are placed near each other, inspired by how fish swim in a school (Figure 12.16), the value of free stream airspeed, U in Equation 12.46, is changed due to the effect that the flutter of the leading cantilevers have on the free stream air by generating vortices in the air flow. Therefore, there is an optimum separation distance between the piezoelectric cantilevers that can maximize the wind energy harvesting (Figure 12.17) by utilizing the vortices in the flow.

Various types of piezoelecteric energy harvesters can be realized depending on the input excitation source and the mechanism that the harvesters use to capture the energy. For instance a windmill-type renewable energy harvester with rotary elements can apply force to a piezoelectric element using a noncontacting excitation mechanism. In such a case, the piezoelecteric element is fixed with a cantilevered-type boundary condition, which is excited by the rotary motion of windmill blades (sails) against it when there are same pole magnetic elements opposing each other on the rotary element and the piezoelectric element. Every time a windmill blade passes against the piezoelectric element, the opposing magnetic force excites the piezoelectric material and therefore generates electricity.

12.4.2 Fish Schooling Inspired Vertical Axis Wind Turbine Farm

A similar scenario as in the bio-inspired piezoelectric energy harvesters (Figure 12.16) is applicable in the design of vertical axis wind turbines (VAWT), which is inspired by fish schooling [64,65]. Bio-inspired spacing between the location of VAWTs investigates the effects of neighboring harvesters on increasing generated power of VAWTs. The modeling of such bio-inspired VAWTs is studied below.

The velocity potential, W, and the velocity vector field, $\boldsymbol{u}$, can be considered as

$$W = U_{\infty} z - i\frac{\Gamma}{2\pi}\log_{10}(z) + \mu z^{-1} \tag{12.51}$$

$$\boldsymbol{u} = \mathrm{Real}\left(\frac{\mathrm{d}W}{\mathrm{d}z}\right)\hat{i} - \mathrm{Imaginary}\left(\frac{\mathrm{d}W}{\mathrm{d}z}\right)\hat{j} \tag{12.52}$$

where Γ denotes the strength of the point vortex, μ is the strength of the dipole, U_{∞} is the free-stream speed. Vectors $\hat{i}$ and $\hat{j}$ represent flow in the x- and y-directions, respectively, at location $z = x + iy$.

For k number of VAWTs, the velocity potential, W can be written as

$$W = U_{\infty} z + \sum_{k=0}^{K}\left[-i\frac{\Gamma}{2\pi}\log_{10}(z - z_k) + \mu(z - z_k)^{-1}\right] \tag{12.53}$$

where z_k denotes the location of the kth VAWT.

The velocity vector that considered the wake effects can be given by [64,79]

$$\boldsymbol{u}^{*}(z) = (1 - \xi\omega(z))\,\boldsymbol{u}(z) \tag{12.54}$$

where $\xi_{\omega}(z)$ is a spatial function representing the velocity deficit.

The power produced by a single isolated VAWT can be determined as

$$P_{\mathrm{iso}} = C_{\mathrm{iso}}\frac{\rho R}{2\pi}\oint_{C}(\boldsymbol{u} * \vec{s})^{3}\,\mathrm{d}s \tag{12.55}$$

Here, ρ is the density of air, C_{iso} is a nondimensional parameter specific to a VAWT, R is the radius of the turbine, $\vec{s}$ is a unit vector, C is the contour of integration, which is a circle with radius of R.

The location of vortices in the wake of schooling fish can be given in the complex plane using Figure 12.16a as [64,80]

$$
\begin{aligned}
A^{1+}_{m,n} &= -a/2 + 2na + i(4mc + b)\\
A^{2+}_{m,n} &= a/2 + 2na + i((4mc + 2)c + b)\\
A^{1-}_{m,n} &= a/2 + 2na + i(4mc - b)\\
A^{2-}_{m,n} &= -a/2 + 2na + i((4m + 2)c - b)
\end{aligned}
\tag{12.56}
$$

Therefore, the velocity potential, *W*, for an array of VAWTs can be obtained by incorporating Equations 12.56 in 12.53, which leads to

$$
\begin{aligned}
W = U_\infty z - i\frac{\Gamma}{2\pi}\sum_{m=1}^{M}\sum_{n=1}^{N}\Big[\log_{10}\left(z - A^{1+}_{m,n}\right)\\
+\log_{10}\left(z - A^{2+}_{m,n}\right) - \log_{10}\left(z - A^{1-}_{m,n}\right) - \log_{10}\left(z - A^{2-}_{m,n}\right)\Big]\\
+\mu\sum_{m=1}^{M}\sum_{n=1}^{N}\Big[\left(z - A^{1+}_{m,n}\right)^{-1} + \left(z - A^{2+}_{m,n}\right)^{-1} + \left(z - A^{1-}_{m,n}\right)^{-1}\\
+\left(z - A^{2-}_{m,n}\right)^{-1}\Big]
\end{aligned}
$$

when this velocity potential is used to calculate the velocity vector field, ***u*** and, therefore, ***u**** (*z*), then the power calculated by this velocity vector corresponds to the power obtained for the VAWTs, which takes into account the effect of vortices in the flow. It has been shown that there is an optimum separation distance for positioning each of the VAWTs relative to each other, which can give more power output compared with a single VAWT in isolation [64].

12.4.3 Bio-Inspired Self-Propelled Vehicle

In this section, a self-propelled vehicle is investigated inspired by jellyfish locomotion. A jellyfish is capable of using vortex formation in the fluid to manipulate the near-wake properties and, therefore, enhance its propulsion efficiency [81–83]. It has been shown that the efficiency of a vehicle that operates with a pulsed-jet propulsion, inspired by jellyfish, can be up to 50% more efficient to that with steady-jet propulsion [82].

The control volume of a vehicle, moving in fluid, is illustrated in Figure 12.19. This control volume is represented by the dashed line in the figure, which has the surface normal of $\boldsymbol{n}_{\mathrm{VCL}}$ and is displaced from the surface of the vehicle by δ.

The model of a vehicle moving in a fluid is depicted in Figure 12.19, in which fluid enters the vehicle at velocity $\boldsymbol{u}_{\mathrm{in}}$ and pressure P_{in} through the vehicle surface as shown, and fluid exists in the vehicle at velocity $\boldsymbol{u}_{\mathrm{out}}$ and pressure P_{out} through a nozzle at the rear.

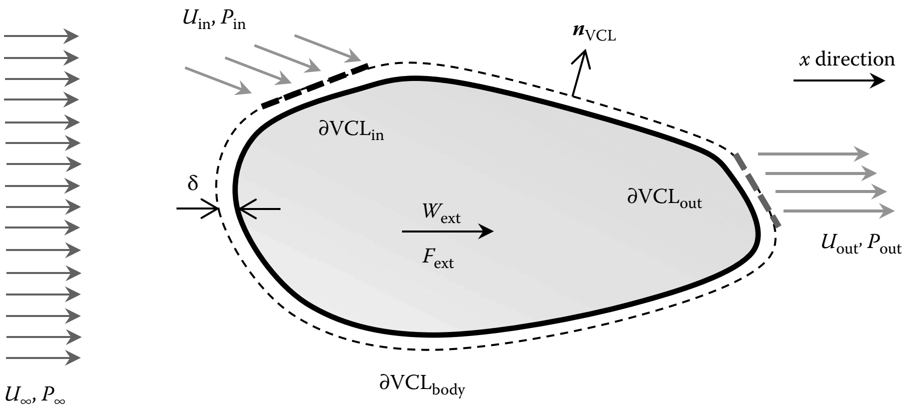


FIGURE 12.19
The vehicle and the associated control volume.

U_∞ is the free-stream flow velocity. From the conversion of mass the following relationship can be written as [82]

$$\rho \oint_{\partial VCL_{out}} \boldsymbol{u}_{out} \cdot \boldsymbol{n}_{VCL}\, dA + \rho \oint_{\partial VCL_{in}} \boldsymbol{u}_{in} \cdot \boldsymbol{n}_{VCL}\, dA = 0 \tag{12.57}$$

The momentum equation in x direction (streamwise) gives

$$F_{ext,x} = \rho \left[\frac{\partial}{\partial t} \int_{VCL} u\, dV + \oint_{\partial VCL_{out}} u_{out}^2\, dA \right] + \oint_{\partial VCL} p n_{VCL,x}\, dA \tag{12.58}$$

where $F_{ext,x}$ denotes the component of the external force vector, $\boldsymbol{F}_{ext}$ (if there is any) in x direction. It is assumed that $|\boldsymbol{u}_{in}| \approx |U_\infty|$ and $P_{in} = P_\infty$.

Steady-jet and pulsed-jet propulsion at a nozzle exit from the vehicle is illustrated in Figure 12.20. Pulsed-jet and the associated control volume boundary of forming wake vortex is depicted in Figure 12.20b.

In each fluid pulse during a pulsed-jet propulsion, the fluid exiting the vehicle forms a large-scale vortex with a well-defined boundary between the "vortex bubble" at the nozzle exit and the ambient fluid [82]. ∂VB represents the vortex boundary with the associated unit normal $\boldsymbol{n}_{VB}$. The inputs to the vortex control volume include the fluid exiting the nozzle and the entrainment mass flux $\dot{m}_{entrain}$. In the presence of the vortex boundary, *VB*, in the pulsed-jet scenario, the output velocity of the nozzle is dependent on the characteristic of the vortex boundary, and therefore, the expression of momentum can be written as

$$F_{ext,x} = \rho \frac{\partial}{\partial t} \int_{VCL \cup VB} u\, dV + \oint_{\partial VCL} p n_{VCL,x}\, dA + \oint_{\partial VB} p n_{VB,x}\, dA \tag{12.59}$$

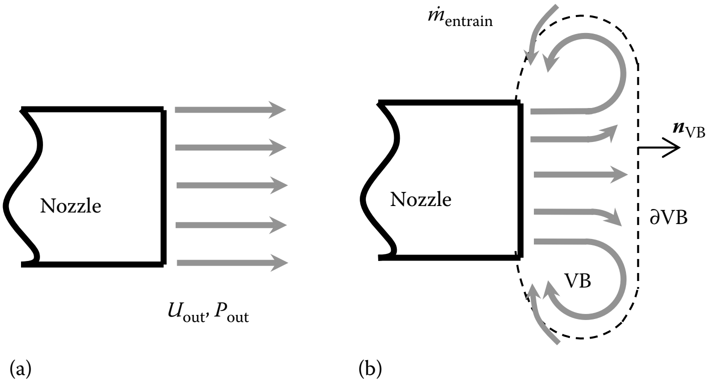


FIGURE 12.20
(a) Steady-jet and (b) pulsed-jet and the vortex wake, and associated control volume.

where VCL ∪ VB denotes the combined vehicle–wake control volume. The pressure integrals in this momentum equation can be stated in terms of the velocity potential ϕ for irrotational flow using the Bernoulli equation as follows.

$$p = \rho\left[C(t) - \frac{\partial\phi}{\partial t} - \frac{1}{2}|\boldsymbol{u}|^2\right] \tag{12.60}$$

where $C(t)$ is the Bernoulli constant.

In Equation 12.59, the flow speed can be approximated as a constant speed $|\boldsymbol{u}| \approx U_\infty$, and $\boldsymbol{n}\mathrm{d}A$ is zero on a closed surface. The added mass can be defined for a streamwise motion as

$$-\oint_{\partial V} \phi n_x \,\mathrm{d}A = \alpha_{xx} V U_x \tag{12.61}$$

where V denotes the volume of ∂V, U_x is the speed of the volume centroid in x direction and α_{xx} is the added-mass coefficient. Considering the added-mass coefficient in Equation 12.59 will lead to

$$F_{\text{ext},x} = \rho\left(1+\alpha_{xx}^{\text{VCL}}\right)\dot{U}_x^{\text{VCL}} V_{\text{VCL}} + \rho\frac{\partial}{\partial t}\left[\left(1+\alpha_{xx}^{\text{VB}}\right)U_x^{\text{VB}} V_{\text{VB}}\right] \tag{12.62}$$

It can be seen from this equation that, in the absence of external forces, the acceleration of the vehicle is dependent on the acceleration of the vortex bubble, which is not the case in a steady-jet propulsion. Therefore the pulsed-jet propulsion is enhanced due to the vortex added mass, which contributes via the added-mass coefficient α_{xx}^{VB} in the analytical formulation (Equation 12.62). The efficiency of a vortex-enhanced propulsion can be given by [82,83]

$$\eta \approx \frac{2}{3}\left(1+\alpha_{xx}^{\text{VB}}\right)\frac{U_\infty}{U_{\text{w}}} \tag{12.63}$$

where U_∞ is the approximate vehicle velocity, and U_w is the approximate wake vortex velocity.

12.4.4 Bio-Inspired Flapping Wing Flying Robots

Birds and insects, unlike air vehicles, are incredibly maneuverable animals, which are capable of rapid changes in their direction of motion [84–90] due to their flapping wing flying system. This maneuverability is desirable for air vehicles. This section studies the dynamics of bio-inspired flapping wing air vehicles, which have potential application in the design and development of unmanned aerial vehicles.

The force applied to the wing in Figure 12.21 can be obtained as [86,87]

$$dF = \frac{1}{2}\rho\dot{\varphi}^2 r^2 C_F(\alpha)c(r)dr \tag{12.64}$$

where F denotes the aerodynamic force applied to the wing, P_dyn is the dynamic pressure, C_F is the force coefficient as a function of angle of attack α, φ is the time varying flapping angle, and $\dot{\varphi}$ is the corresponding angular velocity. r is the radial position along the wing and c is the chord as in the figure.

The integration of the force dF with respect to r gives the instantaneous lift force as

$$F_L = \frac{1}{2}\rho\dot{\varphi}^2 C_L(\alpha)\bar{c}R^3\int_0^1 (\bar{r})^2 c(\bar{r})\,d\bar{r} \tag{12.65}$$

where $r_s = \int_0^1 (\bar{r})^2 c(\bar{r})\,d\bar{r}$. In the parameters $\bar{r} = r/R$ and $c = A/R$, R and A are the wing radius and area. $C_L(\alpha)$ can be determined as

$$C_L(\alpha) = C_{L_\mathrm{max}}\sin(2\alpha) \tag{12.66}$$

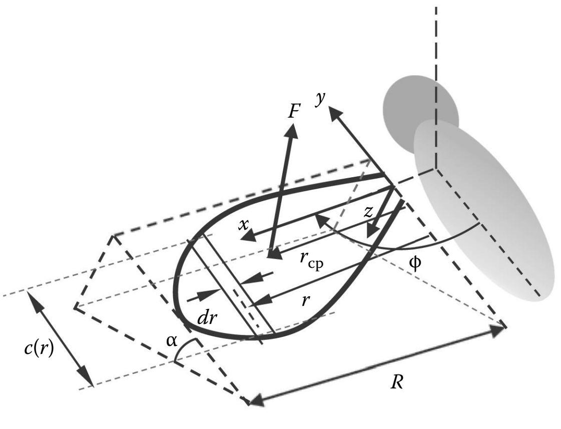


FIGURE 12.21
Parameters of a flapping wing dynamic system.

The drag force, F_D, can be obtained by replacing $C_L(\alpha)$ by $C_D(\alpha)$ in the F_L equation where $C_D(\alpha)$ is

$$C_D(\alpha) = \left(\frac{C_{D_{\max}} + C_{D_0}}{2}\right) - \left(\frac{C_{D_{\max}} - C_{D_0}}{2}\right)\cos(2\alpha) \tag{12.67}$$

$C_{L_{\max}} = 1.8$, $C_{D_{\max}} = 3.4$, and $C_{D_0} = 0.4$ are given by the experimental measurement [88] carried out for a Reynolds number near 200, which varies with small variations up to a Reynolds number of 10,000 [89].

The lift force can be estimated using the assumption of the values given below, which correspond to approximate values of typical insect wings [86,87]:

$$R = 0.01\,\text{m},\ \bar{r}_{\text{cp}} = 0.6,\ R/c = 4,\ r_s = 0.56 \tag{12.68}$$

where $\bar{r}_{\text{cp}} = r_{\text{cp}}/R$ refers to the nondimensional radial position of the center of pressure. The density of air is considered to be $\rho = 1.2\ \text{kgm}^{-3}$. If a change of the angle of attack, α, and the flapping angle, φ, with respect to time is considered as in Figure 12.22, then the lift force, F_L, can be calculated corresponding to this flapping angle using Equations 12.65 and 12.66 and the values given in Equation 12.68.

The lift force, F_L, is plotted versus time in Figure 12.23, when the period of oscillation is considered as 0.01 sec.

The lift force, F_L, can also be plotted in terms of angle of attack, α, and the flapping angle, φ as in Figure 12.24 (period of oscillation is considered as 0.01 sec).

The solid arrow in Figure 12.24 shows how the surface in this figure corresponds to the curve in Figure 12.23 when it is compared with the dashed arrow line in Figure 12.23.

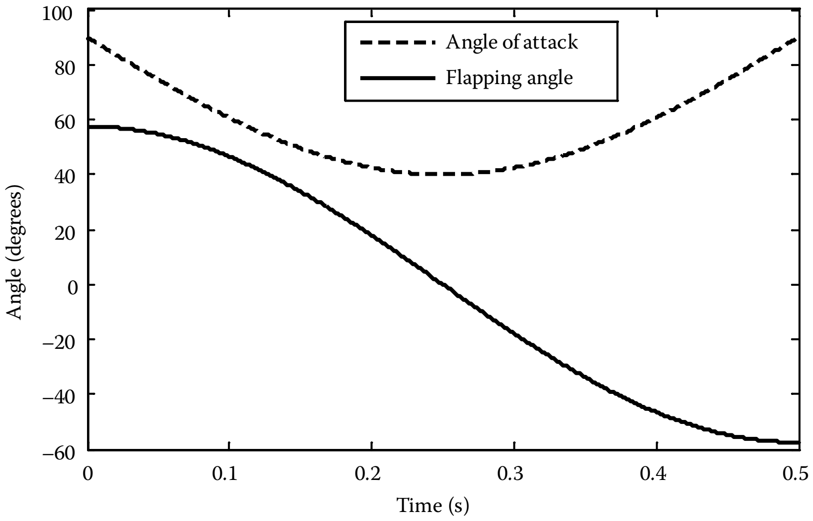


FIGURE 12.22
An example of change of the angle of attack, α, and the flapping angle, φ, with respect to time.

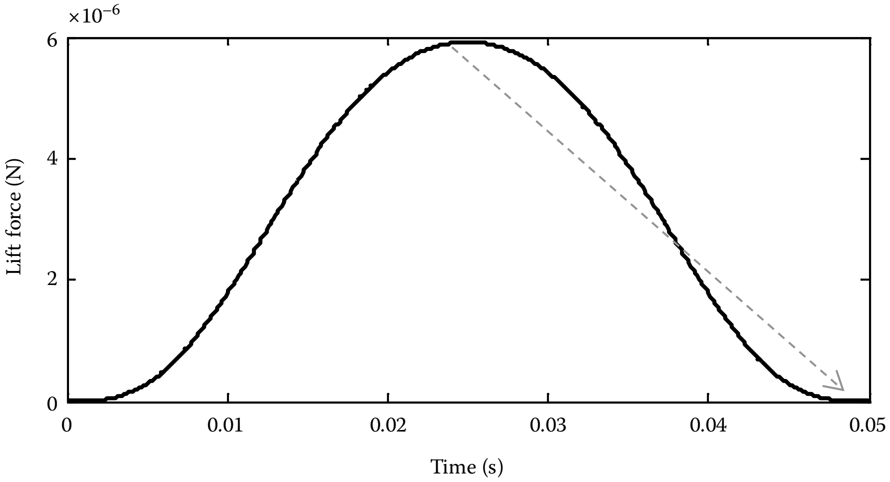


FIGURE 12.23
The lift force, F_L, versus time for the period of oscillation of 0.01 sec.

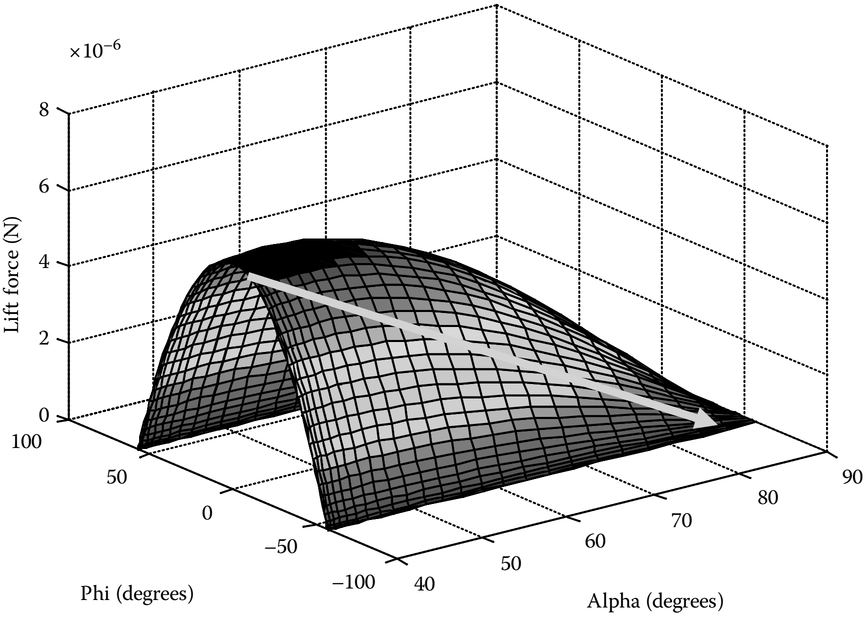


FIGURE 12.24
The lift force, F_L, versus angle of attack, α, and the flapping angle, φ as with the period of oscillation considered as 0.01 sec.

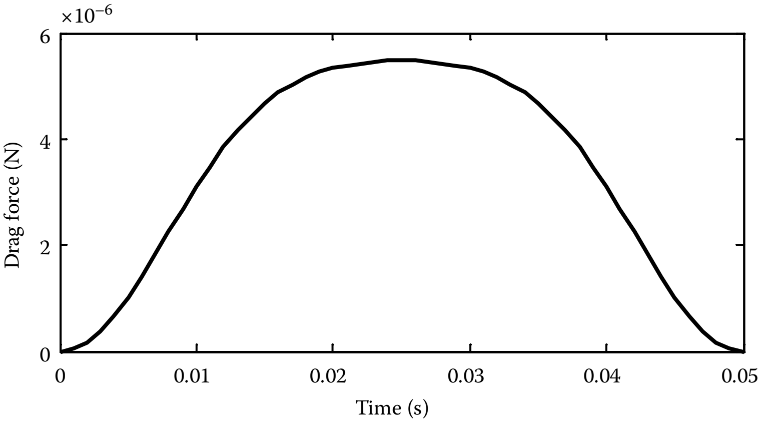


FIGURE 12.25
The drag force, F_D, versus time for the period of oscillation of 0.01 sec.

For the variation of the angle of attack, α, and the flapping angle, φ, given in Figure 12.22, the drag force, F_D, can be calculated using Equations 12.67 and 12.69 and the values given in Equation 12.68.

$$F_D = \frac{1}{2}\rho\dot{\varphi}^2 C_D(\alpha) c R^3 r_s \tag{12.69}$$

where Equation 12.69 is obtained similarly as the lift force in Equation 12.65 except that the lift coefficient is replaced with the drag coefficient, C_D.

The drag force, F_D, is plotted versus time in Figure 12.25, when the period of oscillation is considered as 0.01 sec.

Figure 12.26 represents the lift and drag forces simultanously.

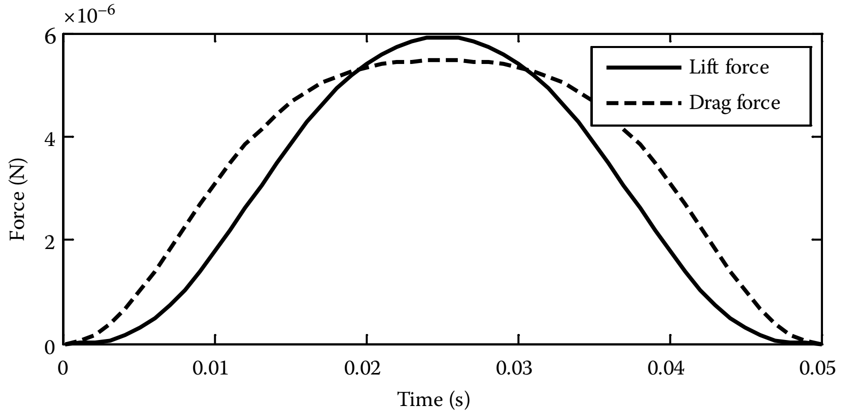


FIGURE 12.26
The lift and drag forces versus time for the period of oscillation of 0.01 sec.

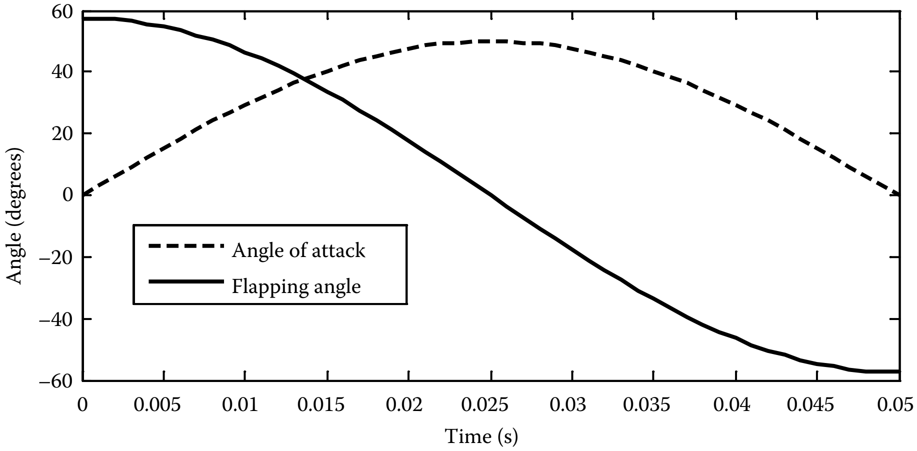


FIGURE 12.27
An example of change of the angle of attack, α, and the flapping angle, φ, with respect to time.

The relative variation of the angle of attack, α, and the flapping angle, φ, may be designed differently from the angles considered as in Figure 12.22. For instance, if the angles are considered as in Figure 12.27, then the lift and the drag forces can be calculated using Equations 12.65 through 12.69, which give the results that are shown in Figure 12.28.

It is observed that for the new set of α and φ given in Figure 12.27, the maximum drag force is higher than the maximum lift force value (Figure 12.28) unlike the previous observation in Figure 12.26 in which the peak lift force is higher. Therefore, the effect of the variation of the flapping angle and the angle of attack on the aerodynamic forces on wings can be investigated as discussed above.

Assuming a linear model, the damping force should be proportional to the velocity, $\dot{\varphi}$. In order to deal with the quadratic term $\dot{\varphi}^2$, the "secant" approximation can be employed

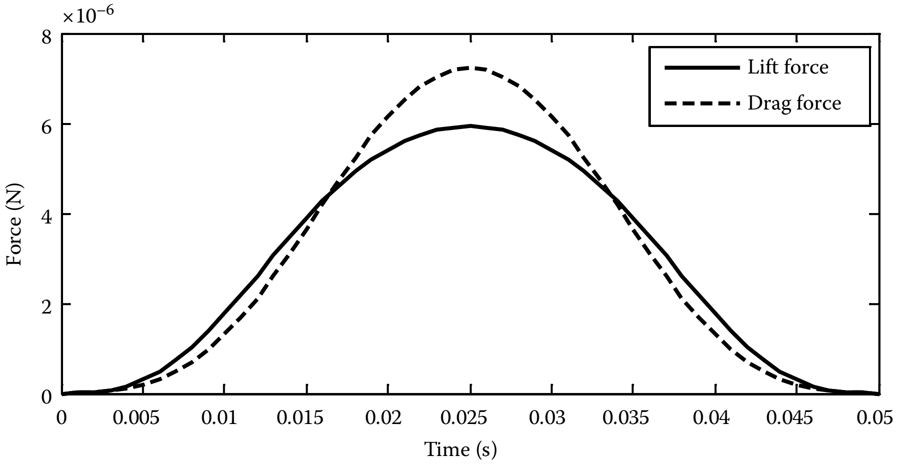


FIGURE 12.28
The lift and drag forces versus time for the period of oscillation of 0.01 sec.

which considers $\dot{\varphi}^2$ as $\dot{\varphi}\dot{\varphi}_0$. In this assumption, $\dot{\varphi}_0$ is the midstroke velocity at $\varphi = 0$. Therefore, if $\varphi = \Phi\sin(\omega t)$, which gives $\dot{\varphi} = \Phi\omega\cos(\omega t)$, for a harmonic motion assumption, then $\dot{\varphi}_0 = \omega\Phi$.

Based on the above assumption and Equation 12.69, the moment applied to the wing can be considered as

$$M_D = \frac{1}{2}\rho\dot{\varphi}\dot{\varphi}_0 C_D(\alpha) c R^3 r_s r_{\text{cp}} \tag{12.70}$$

This can be rewritten in terms of the frequency of the motion, ω, as

$$M_D = \frac{1}{2}\rho\dot{\varphi}\omega\Phi C_D(\alpha) c R^3 r_s r_{\text{cp}} \tag{12.71}$$

If $M_D = b\dot{\varphi}$, where b is the rotational damping, then

$$b = \frac{1}{2}\rho\omega\Phi C_D(\alpha) c R^3 r_s r_{\text{cp}} \tag{12.72}$$

The moment applied by the actuator is required to generate the flapping moment, M_{input}, (Figure 12.29) for each wing as given by $I\dot{\varphi} + M_D = M_{\text{input}}$.

The dynamics of the hovering for a vehicle can be analyzed by obtaining the average lift force, which can generate a force equal to the weight of the vehicle, where the generated force by each wing is half the weight of the vehicle.

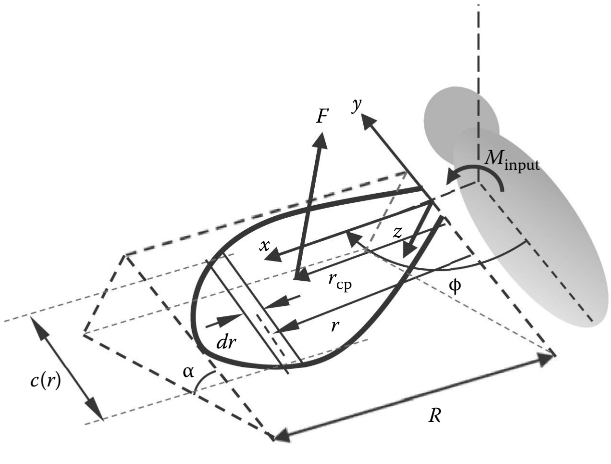


FIGURE 12.29
Actuation moment applied to the wing by the actuator.

The average lift force for hovering can be determined by integrating the lift force in Equation 12.65 over one half-period and using Equation 12.66 for the lift coefficient. Therefore, for a sinusoidal motion, the average lift force can be expressed as

$$L = \frac{1}{2}\rho c R^3 r_s \frac{1}{\pi}\int_0^{\pi} C_{L_{\max}} \sin(2\alpha(x))\Phi^2\omega^2 \cos^2 x \, \mathrm{d}x \tag{12.73}$$

If $\tilde{C}_L = \frac{C_{L_{\max}}}{\pi}\int_0^{\pi} \sin(2\alpha(x))\cos^2 x \, \mathrm{d}x$, then this equation can be rewritten as

$$L = \frac{1}{2}\rho c R^3 r_s \Phi^2 \omega^2 \tilde{C}_L \tag{12.74}$$

If the weight of the vehicle is W, the lift force generated by each wing for hovering is required to be equal to $L = \frac{W}{2}$. Therefore the required flapping frequency for hovering can be obtained by rearranging Equation 12.74 and solving for ω as follows:

$$\omega = \frac{1}{R\Phi}\sqrt{\frac{W}{\rho c R r_s \tilde{C}_L}} \tag{12.75}$$

The numerical value for the hovering frequency can be estimated using the assumption of the values given below, which correspond to approximate values of typical insect wings [86,87]:

$$R = 0.01\text{ m},\ \bar{r}_{\text{cp}} = 0.06,\ R/c = 4,\ r_s = 0.56,\ \tilde{C}_L = 1.8 \tag{12.76}$$

and Φ is the amplitude of the flapping that can be considered as the amplitudes given in Figure 12.22 or Figure 12.27. In Figure 12.30, the generated lift force for hovering is plotted versus the frequency of flapping using Equation 12.75 and the parameter values in Equation 12.76. When using Equation 12.75 for plotting the lift force in Figure 12.30, it is assumed that the required lift force is equal to the vehicle weight W.

The power that can be delivered to a flapping wing vehicle depends on the actuator size. If the actuator mass is m_{a}, and the energy density of the actuator for delivering the power in each flapping period is E, then the actuator size in terms of m_{a} can be obtained using the following relationship.

$$m_{\text{a}}E = L r_{\text{cp}} \Phi \tag{12.77}$$

where the energy density for the insect flight muscle and piezoelectric actuator can be considered as $E = 1.5\text{ J kg}^{-1}$ [90,91]. It should be noted that insects carry carbohydrate or fat energy stores, which has the advantage of much higher energy density than, for instance, lithium batteries in a mechanical vehicle. The energy density of carbohydrate can be approximately 50 times higher than batteries [86]. Therefore, the design of flapping wing flying vehicles is limited to the current energy storage technology.

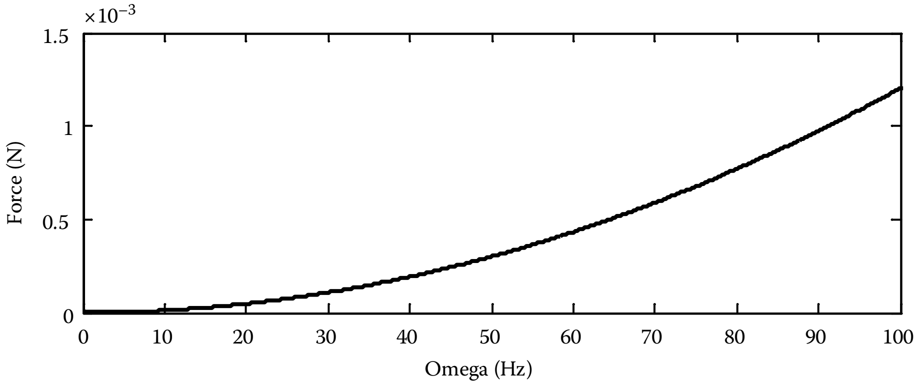


FIGURE 12.30
The generated lift force for hovering versus the frequency of flapping.

By substituting the lift force L from Equation 12.74 into Equation 12.77 then the actuator mass m_a can be obtained as

$$m_\text{a} = \frac{1}{2E}\rho c R^3 r_s \Phi^3 \omega^2 \tilde{C}_L r_\text{cp} \tag{12.78}$$

The actuator mass is part of the total mass of the vehicle, m_T, which needs to be accounted for in the design of the vehicle. The total mass of the vehicle can be considered as $m_\text{T} = m_\text{a} + m_\text{b} + m_\text{r}$ where m_b is the battery mass, and m_r is the mass of the remaining components.

The flight endurance depends on the generated lift force by the power source and the weight of the vehicle. For a battery as the power source with the energy density of E_b, the flight endurance can be determined by

$$t = \frac{E_\text{b} m_\text{b}}{P/\eta} \tag{12.79}$$

where η is the efficiency of the actuator (approximately equal to 10% [90,91]), and P is the power consumption by the flapping wings, and a typical value for E_b for lithium polymer batteries is $E_\text{b} = 500\ \text{J kg}^{-1}$ [92].

12.4.5 Bio-Inspired Flight Control System

As pointed out in the previous section, the maneuverability of birds and insects is greatly superior to air vehicles. One way of designing an air vehicle with improved maneuverability is to mimic the bird and insect flight control mechanism. This includes understanding how birds use visual information to guide their transitions between major modes of flight. An experimental investigation of obtaining the control parameters of birds' visual control for various modes of flight is shown in Figure 12.31 [93]. The key components of flight control of birds' visual control are the brain, eyes, wings, and tail. The brain sends the required nervous signals to the wings and the tail in order to guide the flight.

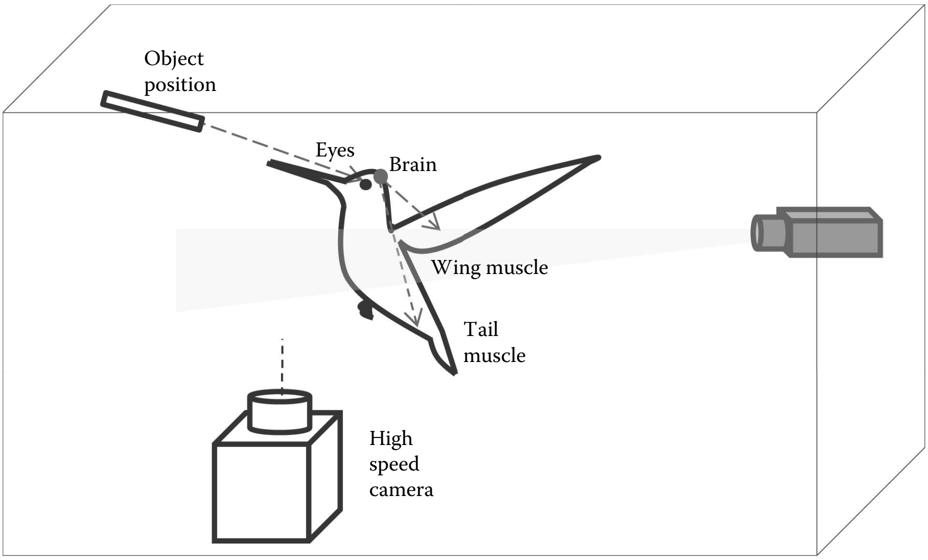

Object position
Eyes
Brain
Wing muscle
Tail muscle
High speed camera

FIGURE 12.31
The experimental setup for obtaining the birds' flight control parameters. (From Altshuler, D. L., Princevac, M., Pan, H., and Lozano, J., *Exp Fluids*, 46, 835–846, 2009.)

The signals command the wings and tail muscles. The muscles act as actuators in a control system (Figure 12.32). The eyes are the sensing components of the system, which continuously provide feedback for the flight control. The position and velocity of the bird are the outputs for the control system. In this control system scenario, the input is the desired position of the bird. The objectives of the control system include minimizing the time and energy to reach the desired position with no steady-state error. Figure 12.32 is an illustration of the control block diagram of birds' position control system. This can represent the control block diagram of the flight control in Figure 12.31. The dynamic system that performs analogous to the birds' flight motion can be modeled as a second-order dynamic system based on Newton's second law of motion.

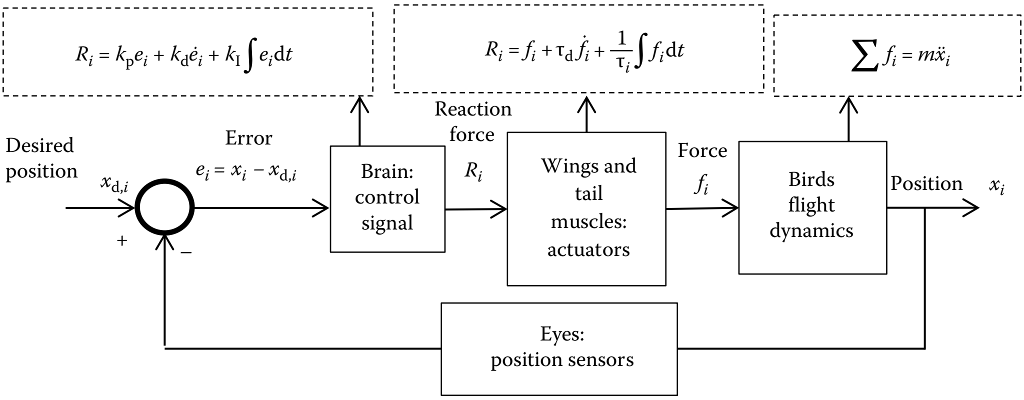

$R_i = k_p e_i + k_d \dot{e}_i + k_I \int e_i \mathrm{d}t$
$R_i = f_i + \tau_d \dot{f}_i + \frac{1}{\tau_i}\int f_i \mathrm{d}t$
$\sum f_i = m\ddot{x}_i$
Desired position $x_{d,i}$
+
−
Error $e_i = x_i - x_{d,i}$
Brain: control signal
Reaction force R_i
Wings and tail muscles: actuators
Force f_i
Birds flight dynamics
Position x_i
Eyes: position sensors

FIGURE 12.32
The bird flight control block diagram.

The parameters represented in the control block diagram in Figure 12.32 are denoted as

R_i = Reaction force required by muscle i

f_i = Force generated by muscle i

e_i = Error between the desired position and current position

x_i = Current position for component i (x_i is linked to the overall position of the bird by a transformation T_i)

$x_{\mathrm{d},i}$ = Desired position for component i ($x_{\mathrm{d},i}$ is linked to the overall position of the bird by a transformation T_i)

m = Mass of the bird

k_p = Proportional controller gain

k_d = Differential controller gain

k_I = Integral controller gain

τ_d = Differential controller time

τ_i = Integral controller time

Using the results obtained in the experimental measurements of the bird flight in Figure 12.31, control parameters can be attained that can be utilized to design a vehicle flight control system. Hence, a controller and actuator system can be designed in order to mimic the controllability of the birds. It should be noted that there are physical limits to the performance of electromechanical actuators, which may not allow achieving the sophistication of animal locomotion.

Visually guided flight and navigation of insects and birds can be used in obtaining equivalent control parameters in development of a bio-inspired control model and system. Examples of such experiments are illustrated in Figure 12.33 [94,95]. A bird is shown in the figure for illustrative purposes although this is a general illustration that can encompass various experimental tests on animal locomotion analysis, particularly insects, in obtaining flight optomotor control response parameters. When an insect is flying along a straight line and a gust of wind forces the insect to move to one side, then the image on its retina senses the change and sends the signal to the brain. The flight motor system generates a corrective yaw torque to return the insect to the straight line again [95,96]. This mechanism is also recognized in pitch and roll motion control. The experimental investigation of understanding the yaw motion control of a flying animal can be realized using a striped rotating drum with which the bird/insect attempts to follow the rotation inside the drum in Figure 12.33a. The visual control of birds/insects can obtain a balance when flying in between narrow gaps from the sensing signals of the two eyes. This can be demonstrated by the experiment in Figure 12.33b and c when an insect/bird flies in between two moving images with various patterns and speeds of motion [94,95].

12.4.5.1 Formation Flight of Birds

There are two main hypotheses associated with the flight of birds in formations. One, it offers aerodynamic advantages, and the other it improves visual communication. It is suggested that tracking the lateral position by birds in formation flight is inherently difficult [97]. The total flight power reduction of a formation flight depends on the lateral distance of the wings. The longitudinal distance of wings does not have any effect on the flight power reduction.

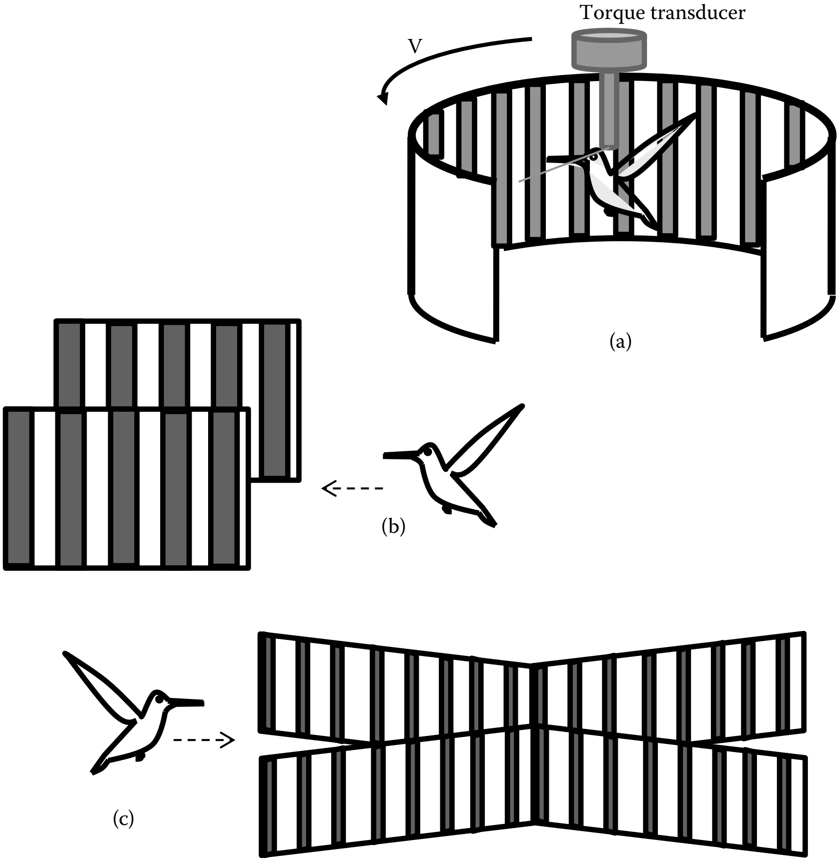


FIGURE 12.33
Experiments for obtaining flight properties of optomotor control response parameters. (a) The experimental investigation of understanding the yaw motion control of flying animal can be realized using a striped rotating drum where the bird/insect attempts to follow the rotation inside the drum. (b and c) The visual control of a bird/insect can obtain a balance when flying in between narrow gaps from the sensing signals of the two eyes when an insect flies in between two moving images with various patterns and speeds of motion.

Study of distance between vortices generated by bird flight in a fixed-wing analysis shows the distance of $\pi L/4$ [97,98] between these vortices (Figure 12.34) where L is the bird wingspan. The following relationship gives the optimum value of wingtip spacing (WTS_{opt}) between birds for receiving maximum lift [97].

$$WTS_{opt} = (\pi L/4 - L)/2 \tag{12.80}$$

Study of aircraft spacing for a two-aircraft formation has an induced-drag reduction of up to 30%, and a three-aircraft formation offers an induced-drag reduction of up to 40% [99].

12.4.6 Uncertainty Quantification

In modeling and simulation of a system, there is an unavoidable level of uncertainty associated with the parameters of the system under investigation. For instance, in the case of a self-powered system, the parameter values for elasticity, damping, energy sources, energy

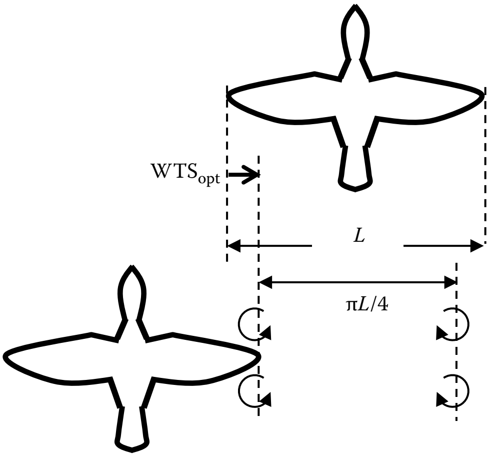


FIGURE 12.34
Formation flight.

conversion mechanisms, energy storage systems, and various electromechanical parameters involved in the system may not be available as exact values, or the value of the parameter may be valid only for a range of variation of system parameters. In the case of a bio-inspired system, depending on the system, uncertainties may arise from various sources, such as the simulation of the fluid, structural parameters of the system and fluid structure interactions, electromechanical properties, etc. Uncertainty quantification (UQ) is a technique that attempts to quantitatively understand the relationships among uncertain parameters and physical processes or in a mathematical model. An optimal uncertainty quantification (OUQ) [100] can be realized as a framework for UQ. The model of a system can be defined in the OUQ framework, which can give optimal bounds on uncertainties. The optimal bounds are values of optimization problems corresponding to extremizing probabilities of deviations from the exact model, subject to the constraints imposed by the assumption and information. Optimal bounds on uncertainties can be defined in terms of the probabilities of deviation from the predefined parameters of the system while imposing the optimization constraints. The constraints for the optimization problem can be defined as the maximum values of the system parameters under investigation. A brief introduction to a simulation problem that deals with uncertainties in modeling in the context of OUQ is presented below using the following example.

Assume $u(t)$ is the displacement vector of a dynamic system defined as

$$u(t) = T\, u_0(t) + v(t) \tag{12.81}$$

where $v(t)$ denotes the oscillation of the dynamic system about the desired position, $u_0(t)$ is the displacement of the coordinate system fixed to the dynamic system, and T is a matrix with arrays of $T_{ij} = 1$ if the ith degree of freedom is a displacement in the jth direction; and $T_{ij} = 0$ otherwise. Substituting Equation 12.81 into the dynamic equation of motion of the system, using Newton's second law, gives

$$M\ddot{v}(t) = \Sigma F(t) - MT\ddot{u}_0(t) \tag{12.82}$$

where $-MT\ddot{u}_0(t)$ can be considered as the effective forces that causes undesirable oscillation of the dynamic system. It is assumed that the dynamic system is required to remain in a certain level of oscillation in order to satisfy stability criterion and certification where the certification condition is considered as

$$\|v\|_\infty < C \tag{12.83}$$

where C is the maximum acceptable displacement deviation from the desired position.

A certification problem in OUQ is defined with constraints on inputs, which correspond to the information set available for input forces. If the input force is characterized by a transfer function ψ, then $\ddot{u}_0(t)$ can be given by

$$\ddot{u}_0(t) := (\psi * w)(t) \tag{12.84}$$

where w is the external excitation velocity vector associated with the external input force with random amplitude. w can be written as

$$w(t) := \sum_{i=1}^{B} X_i w_i(t) \tag{12.85}$$

where $X_1, \ldots, X_B$ are independent random variables with support in $[-\max A, \max A]^3$, where A is the amplitude of the excitation. Components of X_i are $X_{i,1}$, $X_{i,2}$, and $X_{i,3}$. The maximum forces applied to the system can be defied by the following relationship as a function of $A_{\max}$ as

$$F_{\max} = f(A_{\max}) \tag{12.86}$$

The optimal bound on the probability that the dynamic system will deviate from the desired position is the solution of the OUQ problem below [100]. The output optimal bound is obtained by solving the following optimization problem (e.g., using genetic algorithm) with constrained uncertain inputs.

$$\mathcal{U}(\mathcal{A}) := \sup_{(F,\mu)\in\mathcal{A}} \mu[E \le 0] \tag{12.87}$$

where $\mathcal{A}$ is the set of pairs (E, μ) such that (a) E is the mapping of the external force $t \mapsto \ddot{u}_0(t)$ onto the margin $\min\,(C - v)$ by Equations 12.83 through 12.86. (b) μ is the probability measure on the external force excitations defined by Equations 12.84 through 12.86 (where B can be $B = 20$).

A self-powered system can be analyzed in the framework of optimal uncertainty quantification where OUQ allows obtaining the amount of output power in accordance with bounded uncertain inputs [101]. The input energy to a system from a regenerative or renewable source can be introduced as uncertain inputs with lower and upper bounds. The optimal bound on the probability output energy for providing a sufficient energy input required for self-powered operation can be formulated as below.

$$\mathcal{A} := \left\{ (f,\mu) \Big|_{m_1 \le E_\mu[f] \le m_2}^{\mu=\mu_1\otimes\mu_1\otimes\cdots\otimes\mu_n} \right\}$$

where μ_i is the uncertain input i, f is the transfer function of the system, which gives the output value for input μ, E_μ denotes the average output value for the corresponding uncertain inputs μ. m_1 and m_2 are output lower and upper bounds, respectively, if known. The uncertain inputs, μ_i, can be considered as excitation inputs, energy inputs, and/or uncertain system parameters. Note that μ_i inputs are constrained inputs with corresponding lower and upper bounds for each input. The output energy lower and upper bounds (range) is therefore obtained by solving a constrained optimization problem, $\mathcal{U}(\mathcal{A})$, with inputs being constrained: $\mathcal{U}(\mathcal{A}) := \sup_{(F,\mu)\in\mathcal{A}} \mu[E \leq 0]$.

The optimization can be carried out using any optimization techniques, such as genetic algorithm.

12.5 Conclusions

In this chapter, self-powered dynamic systems were reviewed as fully or partially self-powered systems requiring zero or reduced external energy inputs. In such systems, the dynamic system is powered by its own excessive kinetic energy, renewable energy, or a combination of both. The illustrative concept was presented along with the background theory. Various applications and scenarios were addressed in regards to self-powered dynamic systems as self-powered sensors, regenerative actuators, human-powered devices, and solar-powered airships as examples of self-sustained systems. A bio-inspired piezoelectric energy harvester design was investigated to demonstrate the advantage of employing biomimetics to improve power density of an energy harvesting system. Bio-inspired flapping wing flying robots and bio-inspired propulsion were also studied.

References

1. Khoshnoud, F., Y. K. Chen, R. K. Calay, C. W. de Silva, and H. Owhadi, "Self-powered dynamic systems," *European Conference for Aeronautics and Space Sciences*, Munich, Germany, Paper No. 275, July 1–5, 2013.
2. Khoshnoud, F., Y. Zhang, R. Shimura, A. Shahba, G. Jin, G. Pissanidis, Y. K. Chen, and C. W. De Silva, "Energy regeneration from suspension dynamic modes and self-powered actuation," *ASME/IEEE Transaction on Mechatronics*, PP(99), pp. 1–12.
3. Khoshnoud, F., D. B. Sundar, M. N. M. Badi, Y. K. Chen, R. K. Calay, and C. W. de Silva, "Energy harvesting from suspension systems using regenerative force actuators," *International Journal of Vehicle Noise and Vibration*, 9(3/4), pp. 294–311, 2013.
4. Khoshnoud, K., J. Lu, Y. Zhang, R. Folkson, and C. W. De Silva, "Suspension energy regeneration for random excitations and self-powered actuation," *IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, pp. 2549–2554, October 2014.
5. Khoshnoud, F. and C. W. De Silva, "Recent advances in MEMS sensor technology—Mechanical applications," *IEEE Instrumentation and Measurement*, 15(2), pp. 14–24, 2012.
6. Khoshnoud, F., H. Owhadi, C. W. de Silva, W. Zhu, and C. E. Ventura, "Energy harvesting from ambient vibration with a nanotube based oscillator for remote vibration monitoring," *Proc. of the Canadian Congress of Applied Mechanics*, Vancouver, BC, 2011.
7. De Silva, C. W., *Mechatronics—A Foundation Course*, CRC Press/Taylor & Francis, Boca Raton, FL, 2010.

8. Williams, C. B. and R. B. Yates, "Analysis of a micro-electric generator for microsystems," *Sensors and Actuators A*, 52, pp. 8–11, 1996.
9. James, E. P., M. J. Tudor, S. P. Beeby, N. R. Harris, P. Glynne-Jones, J. N. Ross, and N. M. White, "An investigation of self-powered systems for condition monitoring, applications," *Sensors and Actuators A*, 110, pp. 171–176, 2004.
10. Roundy, S., P. K. Wright, and J. Rabaey, "A study of low level vibrations as a power source for wireless sensor nodes," *Computer Communications*, 26, pp. 1131–1144, 2003.
11. Mitcheson, P. D., T. Sterken, C. He, M. Kiziroglou, E. M. Yeatman, and R. Puers, "Electrostatic microgenerators," *Measurement & Control*, 41, pp. 114–119, 2008.
12. Erturk, A. and D. J. Inman, *Piezoelectric Energy Harvesting*, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2011.
13. Mitcheson, P. D. and T. C. Green, "Maximum effectiveness of electrostatic energy harvesters when coupled to interface circuits," *IEEE Transactions on Circuits and Systems. Part 1: Regular Papers*, 59, pp. 3098–3111, 2012.
14. Mitcheson, P. D., T. C. Green, and E. M. Yeatman, "Power processing circuits for electromagnetic, electrostatic and piezoelectric inertial energy scavengers," *Microsystem Technologies-Micro-and Nanosystems-Information Storage and Processing Systems*, 13, pp. 1629–1635, 2007.
15. Muralt, P., M. Marzencki, B. Belgacem, F. Calame, and S. Basrour, "Vibration energy harvesting with PZT micro device," *Procedia Chemistry*, 1, pp. 1191–1194, 2009.
16. Minazara, E., D. Vasic, F. Costa, and G. Poulin, "Piezoelectric diaphragm for vibration energy harvesting," *Proceedings of Ultrasonics International (UI'05) and World Congress on Ultrasonics (WCU)*, 44, pp. e699–e703, 2006.
17. Mitcheson, P. D., T. T. Toh, K. H. Wong, S. G. Burrow, and A. S. Holmes, "Tuning the resonant frequency and damping of an electromagnetic energy harvester using power electronics," *IEEE Transactions on Circuits and Systems. Part 2: Express Briefs*, 58, pp. 792–796, 2011.
18. Ibrahim, S. W. and W. G. Ali, "A review on frequency tuning methods for piezoelectric energy harvesting systems," *Journal of Renewable and Sustainable Energy*, 4(6), 2012.
19. Betts, D. N., H. A. Kim, C. R. Bowen, and D. J. Inman, "Optimal conguration of bistable piezocomposites for energy harvesting," *Applied Physics Letters*, 100, p. 114104, 2012.
20. Mann, B. P. and N. D. Sims, "Energy harvesting from the nonlinear oscillations of magnetic levitation," *Journal of Sound and Vibration*, 319, pp. 515–530, 2009.
21. Stanton, S. C., C. C. McGehee, and B. P. Mann, "Nonlinear dynamics for broadband energy harvesting: Investigation of a bistable piezoelectric inertial generator," *Physica D*, 239, pp. 640–653, 2010.
22. Stephen, N. G., "On energy harvesting from ambient vibration," *Journal of Sound and Vibration*, 293, pp. 409–425, 2006.
23. El-hami, M., P. Glynne-Jones, N. M. White, M. Hill, S. Beeby, E. James, A. D. Brown, and J. N. Ross, "Design and fabrication of a new vibration-based electromechanical power generator," *Sensors and Actuators A: Physical*, 92(1–3), pp. 335–342, 2001.
24. Lee, B., M. A. Rahman, S. Hyun, and G. Chung, "Low frequency driven electromagnetic energy harvester for self-powered system," *Smart Materials and Structures*, 21(12), p. 125024, 2012. DOI:10.1088/0964-1726/21/12/125024.
25. Chao, P. C.–P., "Energy harvesting electronics for vibratory devices in self-powered sensors," *IEEE Sensors Journal*, 11(12), pp. 3106–3121, 2011. DOI:10.1109/JSEN.2011.2167965.
26. Kong, N. and D. S. Ha, "Low-power design of a self-powered piezoelectric energy harvesting system with maximum power point tracking," *IEEE Transactions on Power Electronics*, 27(5), pp. 2298–2308, 2012. DOI:10.1109/TPEL.2011.2172960.
27. Scruggs, J. T., "Structural control using regenerative force actuation networks," PhD Thesis, California Institute of Technology, Pasadena, CA, 2004.
28. Scruggs, J. T. and R. E. Skelton, "Regenerative tensegrity structures for energy harvesting applications," *45th IEEE Conference on Decision and Control*, San Diego, CA, p. 2282, 2006.
29. Nagode, C., M. Ahmadian, and S. Taheri, "Axle generator for freight car electric systems," *Proceedings of the ASME/ASCE/IEEE Joint Rail Conference*, Philadelphia, PA, 2012.

30. Nagode, C., M. Ahmadian, and S. Taheri, "Motion-based energy harvesting devices for railroad applications," *Proceedings of the Joint Rail Conference*, Urbana, IL, 2010.
31. Nagode, C., M. Ahmadian, and S. Taheri, "Vibration-based energy harvesting systems for onboard applications," *Proceedings of the Joint Rail Conference*, Pueblo, CO, 2011.
32. Suda, Y., S. Nakadai, and K. Nakano, "Hybrid suspension, system with skyhook control and energy regeneration (Development of self-powered active suspension)," *Vehicle System Dynamics*, 19, pp. 619–634, 1998.
33. Nakano, K. and Y. Suda, "Combined type self-powered active, vibration control of truck cabins," *Vehicle Systems Dynamics*, 41, pp. 449–473, 2004.
34. Nakano, K., Y. Suda, and S. Nakadai, "Self-powered active vibration control using a single electric actuator," *Journal of Sound and Vibration*, 260, pp. 213–235, 2003.
35. Zuo, L., B. Scully, J. Shestani, and Y. Zhou, "Design and characterization of an electromagnetic energy harvester for vehicle suspensions," *Smart Materials and Structures*, 19, p. 045003 (10 pp.), 2010.
36. Zuo, L. and P. Zhang, "Energy harvesting, ride comfort and road handling of regenerative vehicle suspensions," *ASME Journal of Vibrations and Acoustics*, 135, p. 011002-1, 2012.
37. Karnopp, D., "Permanent magnet linear motors used as variable mechanical dampers for vehicle suspensions," *Vehicle System Dynamics*, 18, pp. 187–200, 1989.
38. Karnopp, D., "Power requirement for vehicle suspension systems," *Vehicle System Dynamics*, 21(1), pp. 65–71, 1992.
39. Segel, L. and X.-P. Lu, "Vehicular resistance to motion as influenced by road roughness and highway alignment," *Australian Road Research*, 12(4), pp. 211–222, 1982.
40. Hsu, P., "Power recovery property of electrical active suspension systems," *Proceedings of the 31st Intersociety Energy Conversion Engineering Conference, (IECEC 96)*, Washington, DC, pp. 1899–1904, August 11–16, 1996.
41. Abouelnour, A. and N. Hamma, "Electric utilization of vehicle damper dissipated energy," *Al-Azhar Engineering Seventh International Conference (AEIC)*, Cairo, Egypt, April 7–10, 2003.
42. Goldner, R., P. Zerigian, and J. Hull, "A preliminary study of energy recovery in vehicles by using regenerative magnetic shock absorbers," *SAE*, Paper No. 2001-01-2071, 2001.
43. Kawamoto, Y., Y. Suda, H. Inoue, and T. Kondo, "Modeling of electromagnetic damper for automobile suspension," *Journal of System Design and Dynamics*, 1, pp. 524–535, 2007.
44. Zhang, Y., K. Huang, F. Yu, Y. Gu, and D. Li, "Experimental verification of energy-regenerative feasibility for an automotive electrical suspension system," *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, Beijing, China, December 13–15, 2007.
45. Chen, Y.-Y., D. Vasic, F. Costa, C.-K. Lee, and W.-J. Wu, "Self-powered semi-passive piezoelectric structural damping based on zero-velocity crossing detection," *Smart Materials and Structures*, 22(2), p. 025029, 2013, DOI:10.1088/0964-1726/22/2/025029.
46. Takeuchi, S., K. Makihara, and J. Onoda, "Reliable and evolvable vibration suppression by self-powered digital vibration control," *Journal of Vibration and Acoustics-Transactions of the ASME*, 134(2), 2012.
47. Margolis, D., "Energy regenerative actuator for motion control with application to fluid power systems," *Journal of Dynamic Systems, Measurement, and Control*, 127(1), pp. 33–40, 2005.
48. Jolly, M. R. and D. L. Margolis, "Assessing the potential for energy regeneration in dynamic subsystems," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 119(2), pp. 265–270, 1997.
49. Jolly, M. R. and D. L. Margolis, "Regenerative systems for vibration control," *Journal of Vibration and Acoustics, Transactions of the ASME*, 119(2), pp. 208–215, 1997.
50. Shen, W.-A., S. Zhu, and Y.-L. Xu, "An experimental study on self-powered vibration control and monitoring system using electromagnetic TMD and wireless sensors," *Sensors and Actuators A-Physical*, 180, pp. 166–176, 2012.
51. Sapinski, B., "Experimental study of a self-powered and sensing MR-damper-based vibration control system," *Smart Materials & Structures*, 20(10), p. 105007, 2011, DOI:10.1088/0964-1726/20/10/105007.
52. Ye, G. and K. Soga, "Energy harvesting from water distribution systems," *Journal of Energy Engineering-ASCE*, 138(1), pp. 7–17, 2012.

53. Cunefare, K. A., E. A. Skow, A. Erturk, J. Savor, N. Verma, and M. R. Cacan, "Energy harvesting from hydraulic pressure fluctuations," *Smart Materials and Structures*, 22(2), p. 025036, 2013, DOI:10.1088/0964-1726/22/2/025036.
54. Shenck, N. S. and J. A. Paradiso, "Energy scavenging with shoe-mounted piezoelectrics," *Micro, IEEE*, 21, pp. 30–42, 2001.
55. Beal, D. N., F. S. Hover, M. S. Triantafyllou, J. C. Liao, and G. V. Lauder, "Passive propulsion in vortex wakes," *Journal of Fluid Mechanics*, 549, pp. 385–402, 2006.
56. Zhu, Q., "Optimal frequency for flow energy harvesting of a flapping foil," *Journal of Fluid Mechanics*, 675, pp. 495–517, 2011.
57. Eldredge, J. D. and D. Pisani, "Passive locomotion of a simple articulated fish-like system in the wake of an obstacle," *Journal of Fluid Mechanics*, 607, pp. 279–288, 2008.
58. Bose, N. and J. Lien, "Energy absorption from ocean waves: A free ride for cetaceans," *Proceedings of the Royal Society of London*, B240, pp. 591–605, 1990.
59. Liao, J. C., D. N. Beal, G. V. Lauder, and M. S. Triantafyllou, "The Karman gait: Novel body kinematics of rainbow trout swimming in a vortex street," *Journal of Experimental Biology*, 206, pp. 1059–1073, 2003.
60. Shimura, R., "Piezoelectric energy harvesting and investigation of bio-inspired enhancement of power density," M.Sc. project, University of Hertfordshire, 2012.
61. Shahba, A., "Investigation of the wind power from underground and tunnels using piezoelectric energy harvesters and a bio-inspired design," M.Sc. project, University of Hertfordshire, 2012.
62. Riaz, O., R. Shah, B. Patel, F. Khoshnoud, G. Pissanidis, Y. K. Chen, R. Shimura, A. Shahba, and G. Gaviraghi, "Piezoelectric energy harvesting for airships and investigation of bio-inspired energy harvesters," *European Conference for Aeronautics and Space Sciences*, Munich, Germany, 2013.
63. Bryant, M., R. L. Mahtani, and E. Garcia, "Wake synergies enhance performance in aeroelastic vibration energy harvesting," *Journal of Intelligent Material Systems and Structures*, 2012.
64. Whittlesey, R. W., S. Lisk, and J. O. Dabiri, "Fish schooling as a basis for vertical axis wind turbine farm design," *Bioinspiration and Biomimetics*, 5, p. 035005 (6 pp.), pp. 1131–1141, 2010.
65. Dabiri, J. O., "Potential order-of-magnitude enhancement of wind farm power density via counter-rotating vertical-axis wind turbine arrays," *Journal of Renewable and Sustainable Energy*, 3, p. 043104, 2011.
66. De Silva, C. W., *Vibration—Fundamentals and Practice*, 2nd ed., Taylor & Francis/CRC, 2007.
67. De Silva, C. W., *Modeling and Control of Engineering Systems*, CRC Press/Taylor & Francis, Boca Raton, FL, 2009.
68. Khoshnoud, F., Y. K. Chen, and R. K. Calay, "On power and control systems of multibody advanced airship for transport," *International Journal of Modelling, Identification and Control*, 18(4), pp. 313–322, 2013.
69. Dumas, A., M. Madonia, I. Giuliani, and M. Trancossi, "MAAT Cruiser/Feeder project: Criticalities and solution guidelines," *SAE Aerotech Congress & Exposition*, Toulouse, October 2011.
70. Dumas, A., M. Madonia, I. Giuliani, and M. Trancossi, "MAAT, multibody advanced airship for transport," *SAE Aerotech, Congress & Exposition*, Toulouse, October 2011.
71. Khoshnoud, F., Y. K. Chen, and R. K. Calay, "An integrated solar-fuel cell powered system for airships," *Proceedings of the IEEE International Conference on Modelling, Identification and Control*, Wuhan, China, June 2012.
72. Available at http://www.eumaat.info.
73. Duffie, J. A. and W. A. Beckman, *Solar Engineering of Thermal Processes*, 2nd ed., John Wiley & Sons Inc., New York, 1991.
74. Nelson, J., *The Physics of Solar Cells*, Imperial College Press, London, 2003.
75. De Soto, W., S. A. Klein, and W. A. Beckman, "Improvement and validation of a model for photovoltaic array performance," *Solar Energy*, 80, pp. 78–88, 2006.
76. Khoury, G. A. and J. D. Gillett, *Airship Technology*, Cambridge Aerospace Series, Cambridge University Press, New York, 1999.

77. Hou, T.-C., Y. Yang, H. Zhang, J. Chen, L.-J. Chen, and Z. L. Wang, "Triboelectric nanogenerator built inside shoe insole for harvesting walking energy," *Nano Energy*, 2(5), pp. 856–862, 2013.
78. Cross, R., "Standing, walking, running, and jumping on a force plate," *American Journal of Physics*, 67(4), pp. 304–309, 1999.
79. Hau, E., *Wind Turbines*, 2nd ed., Springer, New York, 2006.
80. Weihs, D., "Some hydrodynamical aspects of fish schooling swimming and flying in nature." In: *Swimming and Flying in Nature*, eds. T. Wu, C. Brokaw, and C. Brennen, Plenum, New York, vol. 2, pp. 703–718, 1975.
81. Dabiri, J. O., "Optimal vortex formation as a unifying principle in biological propulsion," *Annual Review of Fluid Mechanics*, 41, pp. 17–33, 2009.
82. Ruiz, L. A., R. W. Whittlesey, and J. O. Dabiri, "Vortex-enhanced propulsion," *Journal of Fluid Mechanics*, 668, pp. 5–32, 2011.
83. Whittlesey, R. W. and Dabiri, J. O., "Optimal vortex formation in a self-propelled vehicle," *Journal of Fluid Mechanics*, 737, pp. 78–104, 2013.
84. Karaman, S. and E. Frazzoli, "High-speed flight in an Ergodic Forest," *IEEE Conference on Robotics and Automation*, St Paul, MN, 2012.
85. Willis, D. J., J. Peraire, M. Drela, and J. K. White, "A numerical exploration of parameter dependence in power optimal flapping flight," presented at AIAA Conference, AIAA 2006-2994, San Francisco, CA, June 2006.
86. Whitney, J. P. and R. J. Wood, "Conceptual design of flapping-wing micro air vehicles," *Bioinspiration and Biomimetics*, 7, p. 036001 (10 pp.), 2012.
87. Ellington, C. P., "The aerodynamics of insect flight: II. Morphological parameters," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 305, 17–40, 1984.
88. Dickinson, M. H., F.-O. Lehmann, and S. P. Sane, "Wing rotation and the aerodynamic basis of insect flight," *Science*, 284, pp. 1954–1960, 1999.
89. Lentink, D. and M. H. Dickinson, "Rotational accelerations stabilize leading edge vortices on revolving fly wings," *Journal of Experimental Biology*, 212, pp. 2705–2719, 2009.
90. Dudley, R., *The Biomechanics of Insect Flight: Form, Function, Evolution*, Princeton University Press, Princeton, NJ, 2000.
91. Wood, R. J., E. Steltz, and R. S. Fearing, "Nonlinear performance limits for high energy density piezoelectric bending actuators," *ICRA: Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3633–3640, 2005.
92. Karpelson, M., J. P. Whitney, G. Y. Wei, and R. J. Wood, "Energetics of flapping-wing robotic insects: Towards autonomous hovering flight," *IROS: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1630–1637, 2010.
93. Altshuler, D. L., M. Princevac, H. Pan, and J. Lozano, "Wake patterns of the wings and tail of hovering hummingbirds," *Experiments in Fluids*, 46, pp. 835–846, 2009.
94. Srinivasan, M. V. and S. Zhang, "Visual control of honeybee flight." In: *Orientation and Communication in Arthropods*, ed. M. Lehrer, Birkhauser Verlag, Berlin, pp. 95–114, 1997.
95. Srinivasan, M. V., "Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics," *Physiological Reviews*, 91, pp. 413–460, 2011.
96. Reichardt, W., "Movement perception in insect." In: *Processing of Optical Data by Organisms and by Machines*, ed. W. Reichardt, Academic, New York, pp. 465–493, 1969.
97. Seiler, P., A. Pant, and J. K. Hedrick, "A systems interpretation for observations of bird V-formations," *Journal of Theoretical Biology*, 221, pp. 279–287, 2003.
98. Hummel, D., "Aerodynamic aspects of formation flight in birds," *Journal of Theoretical Biology*, 104(3), pp. 321–347, 1983.
99. Ning, A., T. C. Flanzer, and I. M. Kroo, "Aerodynamic performance of extended formation flight," *Journal of Aircraft*, 48(3), pp. 855–865, 2011.
100. Owhadi, H., C. Scovel, T. Sullivan, M. McKerns, and M. Ortiz, "Optimal uncertainty quantification," *SIAM Review*, 55(2), pp. 271–345, 2013.
101. Khoshnoud, F., H. Owhadi, C. W. de Silva, and I. I. Esat, "Self-powered dynamic systems in the framework of optimal uncertainty quantification," submitting.

13

Visual Servo Systems for Mobile Robots

Haoxiang Lang and Clarence W. de Silva

CONTENTS

13.1 Introduction 499
13.2 Mobile Robotic Visual Servo Systems 501
  13.2.1 State of the Art of Mobile Robotic Systems 501
  13.2.2 Typical Sensors 504
13.3 Visual Servoing 507
  13.3.1 Basic Categories of Visual Servoing 507
  13.3.2 Modeling of Visual Servo System 508
    13.3.2.1 Kinematic Modeling of Robots 508
    13.3.2.2 Camera Modeling 510
    13.3.2.3 Camera Parameters 513
13.4 Case Study of Visual Servoing 514
  13.4.1 System Modeling 514
  13.4.2 Traditional Image-Based Visual Servoing 517
  13.4.3 Adaptive Nonlinear Model Predictive Control 518
13.5 Conclusions 527
References 527

SUMMARY This chapter gives a basic treatment of the problem of visual servoing, especially for mobile manipulation systems. It is a mechatronics problem. The chapter presents an overview of visual servo research, kinematic modeling of robotic systems, camera modeling, and controller design for visual servo systems. Two case studies are given. The first case study presents the development and results of traditional image-based visual servoing. The second case study demonstrates a more advanced controller design of image-based visual servoing. Experimental results are presented for both case studies.

13.1 Introduction

A vision-based mobile manipulation system, also called a visual servo system or visual servoing, refers to the use of computer vision data to determine the sensory feedback for controlling the motion of a robot. It involves fusion of many related research areas including machine vision, robot modeling, control theory, and real-time computing. Specifically, in visual servoing, the vision system provides feedback information about the current state of the process (including its environment) to the controller [1].

The earliest research of visual servoing was reported in 1979 [2]. However, then the image processing procedure took seconds to execute due to the limitation of the computers and image-sensing devices at the time, making real-time control virtually nonfeasible. Sanderson and Weiss (1980) introduced a taxonomy of visual servo systems through a control structure [3]. It is called a dynamic look-and-move structure within which the robotic control architecture is hierarchical and uses vision to provide set point inputs to the joint-level controller. Then the subcontrol system utilizes joint feedback to internally stabilize the robot (Figure 13.1). In contrast, direct visual servoing utilizes a visual servo controller, which directly relies on vision information to compute the joint inputs, thereby stabilizing the robot (Figure 13.2).

With the rapid advancement of computer technologies and image-sensing hardware (CCD and CMOS), computer vision is much faster now than it was in the 1980s. Direct visual servoing came to the attention of researchers in the 1990s. Hutchinson has reviewed much of the related work in [4]. Since then, the term "visual servoing" has come to be accepted as a generic description for any type of visual "feedback control" of a robotic system. The subject has been under study in various forms for more than 20 years, in contexts ranging from simple pick-and-place tasks to today's real-time, complex tasks involving multiple robots and objects, autonomous cooperation, and dynamic, unstructured and unknown environments.

Another major classification of vision-based servoing distinguishes between the position-based approach and the image-based approach [1]. Both approaches share a similar control block diagram with a slight difference in the control feedback loop and the reference input (Figure 13.3). In the former, features are extracted from images from one or more cameras and used in conjunction with camera models and a geometric model of the target object to estimate the pose of the target with respect to the cameras. The controller seeks to reduce the error between the current pose and the desired pose in a three-dimensional (3-D) workspace. In contrast, image-based visual servo control uses 2-D images (and their visible

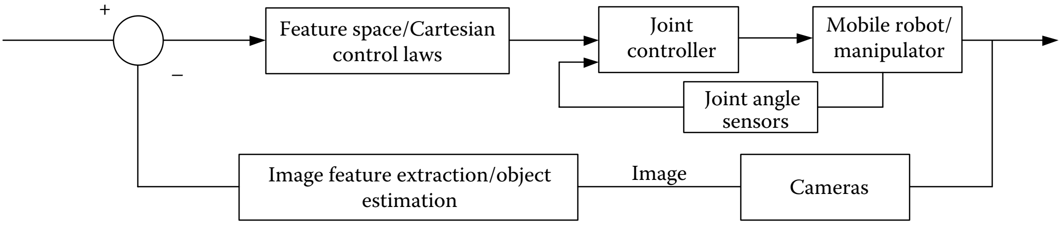

+
–
Feature space/Cartesian control laws
Joint controller
Mobile robot/ manipulator
Joint angle sensors
Image feature extraction/object estimation
Image
Cameras

FIGURE 13.1
Block diagram of a dynamic look-and-move system.

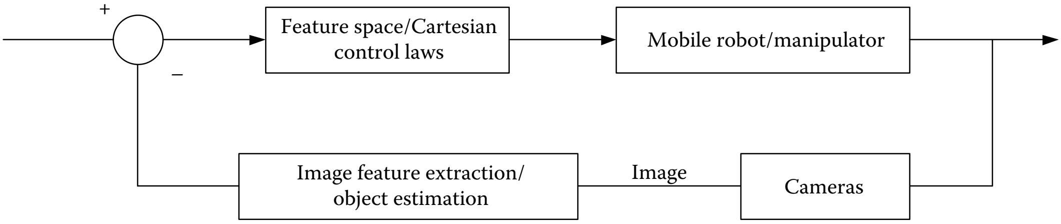

+
–
Feature space/Cartesian control laws
Mobile robot/manipulator
Image feature extraction/ object estimation
Image
Cameras

FIGURE 13.2
Block diagram of a typical (direct) visual servo system.

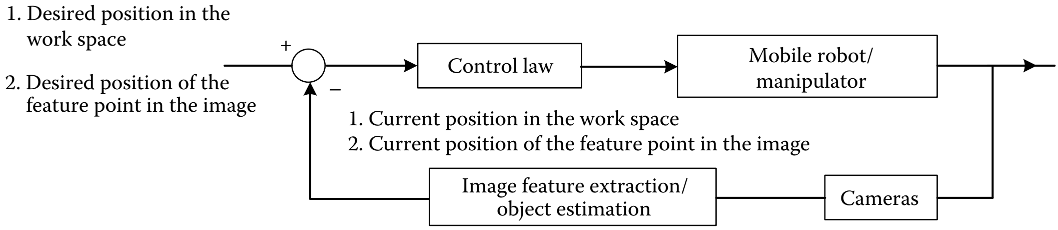

1. Desired position in the work space
2. Desired position of the feature point in the image
+
−
Control law
Mobile robot/ manipulator
1. Current position in the work space
2. Current position of the feature point in the image
Image feature extraction/ object estimation
Cameras

FIGURE 13.3
Block diagram of a typical visual servo system: (1) position-based and (2) image-based.

feature points) directly. Consequently, image-based visual servo control reduces the computational burden, omits unnecessary image interpretation, and eliminates the calibration errors in sensors and cameras.

13.2 Mobile Robotic Visual Servo Systems

Artificial vision is a powerful sensor as it seeks to mimic the human sense of vision and allows noncontact measurement of a work environment. Moreover, it can generate a variety of sensory information, such as position, distance, and features needed for object identification. Because of its advantages, professionals in the robotics field have paid attention to computer vision and practical applications. Since the invention of the camera, much effort has gone into applying vision as a feedback sensor in industrial control systems. Vision-guided robotics, which integrates a vision system into a robot system, has been a major research areas for decades. A common application in this field is the inspection of manufactured goods, such as semiconductor chips, automobiles, food products, and pharmaceuticals. Other recent applications in this area involve vision-based object detection, monitoring of unmanned ground vehicles (UGVs), home care robotics, human rescue robotics, tracking of mobile robots, and vision-based feedback control of robotic manipulator movements (visual servoing).

13.2.1 State of the Art of Mobile Robotic Systems

Among the projects in the application of vision-guided robotics, well known is the DARPA Urban Challenge. This project promoted the development of autonomous mobile robots for the specific purpose of autonomous driving of a vehicle in an urban environment. These robots were designed and developed to adapt to drive in urban traffic while performing complex maneuvers, such as merging, passing, parking, and negotiating intersections. Figure 13.4 shows some example participants in the DARPA Urban Challenge. In these vehicles, the camera is a primary sensor that provides feedback from the environment to the vehicle control system. In [5,6], the details of design and development of the hardware and software for such vehicles were introduced and discussed.

Vision-based automated manipulation has been playing a significant role in industrial and service applications for years. Many of the related applications have been reviewed in [7]. One shortcoming was that it considered fixed-base robotic manipulators only, which are typically applied in structured and known work environments. The new trend is to

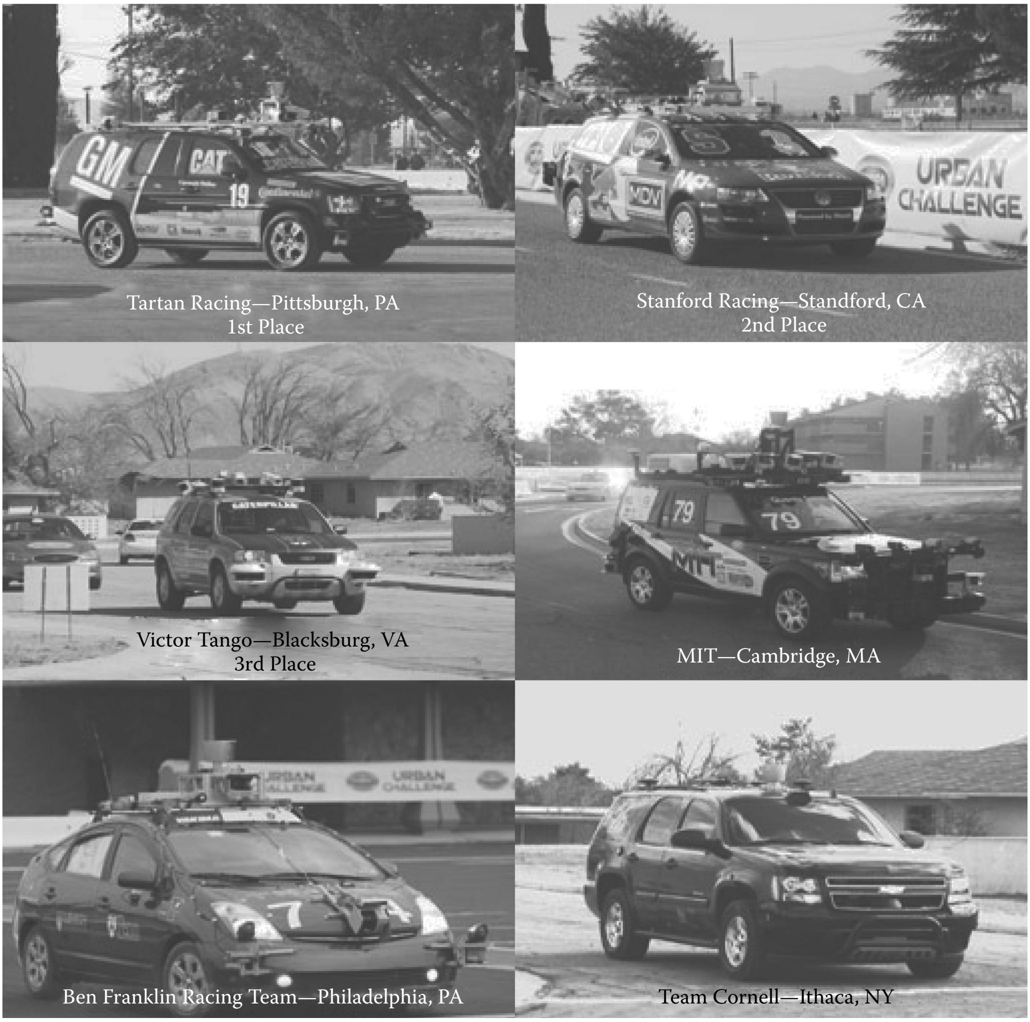

Tartan Racing—Pittsburgh, PA
1st Place
Stanford Racing—Standford, CA
2nd Place
Victor Tango—Blacksburg, VA
3rd Place
MIT—Cambridge, MA
Ben Franklin Racing Team—Philadelphia, PA
Team Cornell—Ithaca, NY

FIGURE 13.4
Some example vehicles in the DARPA Urban Challenge.

integrate visual servoing into a mobile robot for autonomously carrying out the activities of grasping or manipulation, resulting in a vision-based autonomous mobile manipulation system. Compared to traditional visual servo control that is employed in fixed-base robotic manipulators, a vision-based mobile manipulation system has many advantages. Obvious among them is that the robotic system becomes more flexible and has wider applications than a traditional fixed-base manipulator system. Because the robotic arm (manipulator) and the cameras are usually mounted on a mobile base, a mobile manipulation system possesses better maneuverability and terrain coverage capability than a fixed-base manipulator. The historical robotic systems that have been developed since the 1980s are reviewed in [8]. In general, these robotic systems have the conventional architecture of four subsystems: the mobile platform, robotic manipulator (with arm and hand), tooling, and sensors.

Figure 13.5 shows several newly developed mobile manipulation systems for different applications. They are Willow Garage PR2 [9,10] and DLR Rollin' Justin [11] of home care

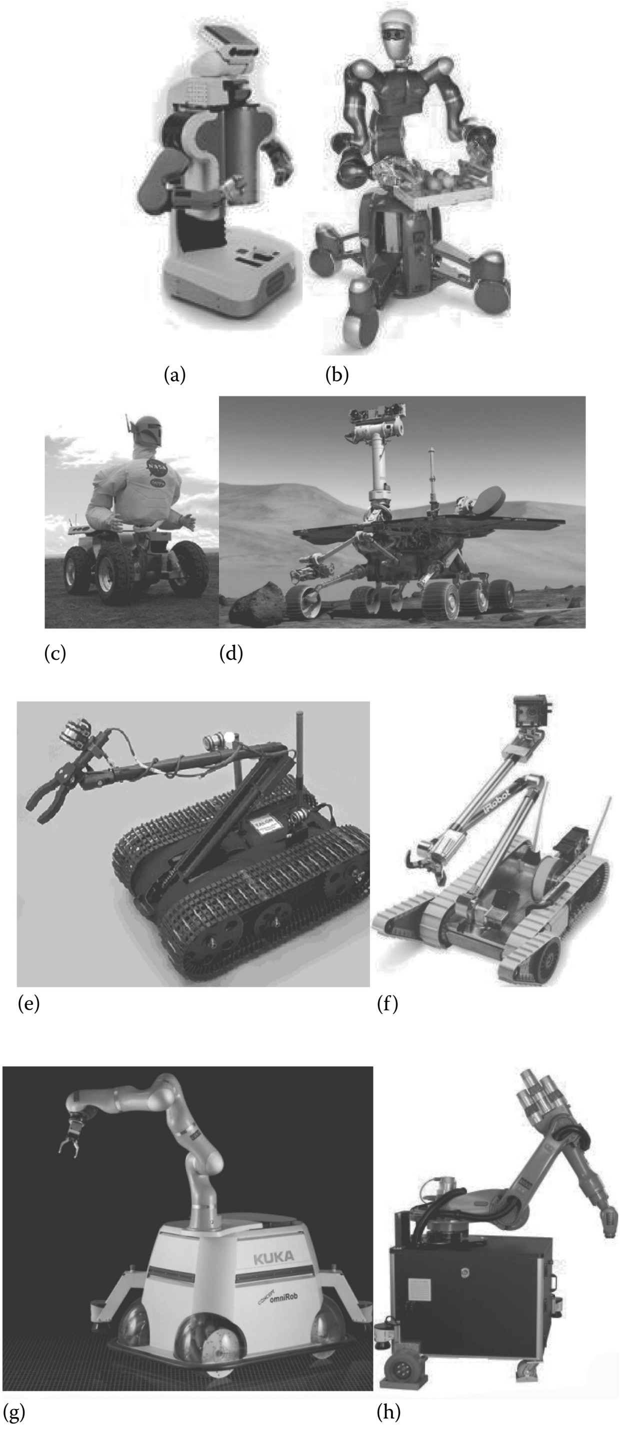

(a)
(b)
NASA
(c)
(d)
TALON
(e)
iRobot
(f)
KUKA
omniRob
(g)
(h)

FIGURE 13.5
Examples of mobile robot manipulation systems in different application areas. Service robots: (a) Willow Garage PR2 and (b) DLR Rollin′ Justin. Space robots: (c) NASA/GM Robonaut and (d) NASA Mars Rover. Military robots: (e) TALON IV engineer and (f) iRobot-PackBot EOD. Industrial robots: (g) KUKA OmniRob and (h) Neobotix.

and service robots; NASA/GM Robonaut [12] and NASA Mars Rover [13] for space robotics; TALON IV engineer [14] and iRobot-PackBot EOD [15], which are applied in military; and KUKA OmniRob [16] and Neobotix [17] in applications of industrial robotics.

13.2.2 Typical Sensors

Many off-the-shelf sensors (e.g., GPS, compasses, gyroscopes, and ultrasonic sensors) that are available for mobile robots are introduced in [18] and [19], giving their operating principles and performance limitations. The camera and laser range finder are the most commonly used sensors in mobile manipulation systems. A vision system is passive and has high resolution. With the help of advanced machine vision technologies, it can provide the location information of objects in the work space using the vision information acquired from the camera scenes. It appears to be the most promising sensor category for future generations of mobile robots.

Two general camera configurations are used in visual servo systems: the eye-to-hand configuration and camera-in-hand configuration, which are shown in Figure 13.6. In the eye-to-hand configuration (Figure 13.6a), the camera is fixed in the work space and observes the robotic "hand." Therefore, the field of view of the camera does not change. The camera keeps tracking the positions of the robot end effector and the target object both, in the image plane, during the visual servo operation. In this configuration, the geometric relationship between the camera, robot, and target object is fixed, the advantage of which is straightforward environmental modeling and 3-D reconstruction. However, the end-effector can easily occlude the camera field of view and block the target object in the image plane. Moreover, because the camera is fixed in the environment, its field of view cannot be changed, which is a limitation in mobile manipulation applications.

In the eye-in-hand configuration, the camera is mounted at the end effector of the robot. In this configuration, the camera will move with the robot. Therefore, the field of view of the camera changes constantly, and it depends on the position and the orientation of the end effector, which is an advantage. It is suitable in applications of visual servoing

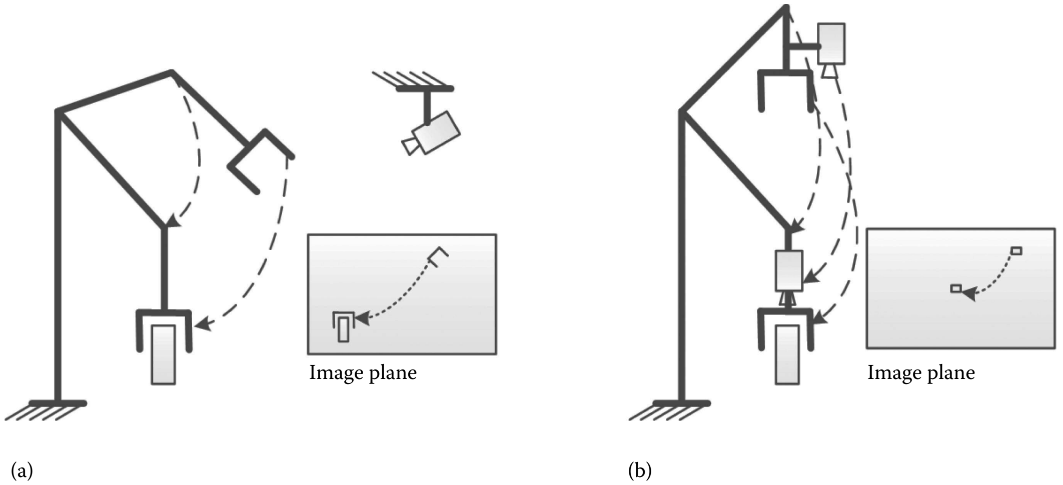


FIGURE 13.6
(a) Eye-to-hand configuration and (b) camera-in-hand configuration.

of mobile robots. Moreover, it can avoid the problem of occlusion because the camera is mounted on the end effector of the robot. However, it has disadvantages as well. For example, the geometry of the camera, robot, and the target object changes as the end effector moves, and the camera view may dramatically change even for small movements of the robot.

In a mobile robot, the base frame is not fixed, and the base coordinate frame changes with time as the robot navigates. A predefined approach to manipulation, as typically used in industry, will not work in this case. Therefore, a camera that is fixed in the environment, which works as an eye-to-hand configuration, is not suitable because it is typically not possible to cover the entire work space of the mobile robot by using a single camera. Therefore, the eye-in-hand configuration is more suitable for mobile manipulation applications.

In addition to the vision information, the distance information is also crucial in mobile manipulation. Sonar and laser distance finders are two typical sensors that are used for sensing distance. Sonar is fast and inexpensive but is usually rather crude whereas laser scanning is active, accurate, and widely applied in mobile robotics. Figure 13.7 shows two types of laser distance finders and their sensing results. The Hokuyo URG-04LX sensor

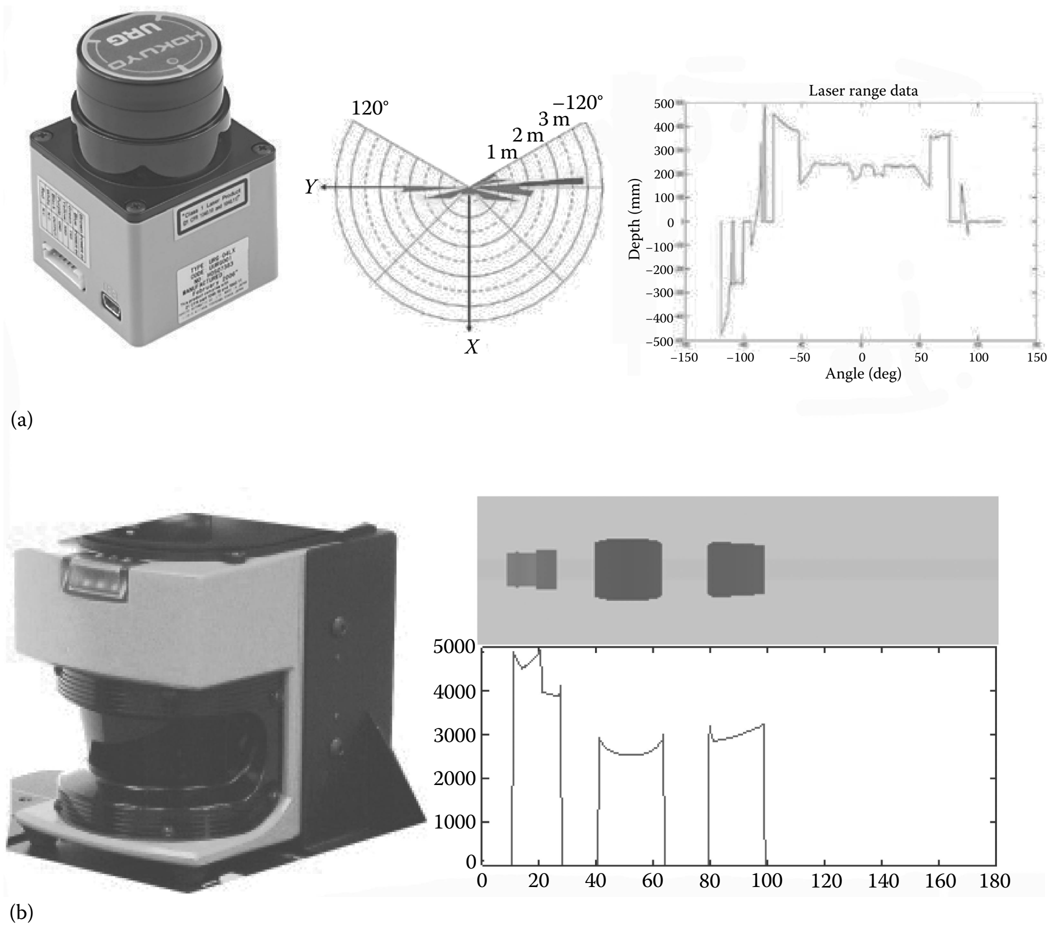

120°
−120°
3 m
2 m
1 m
Y
X
Laser range data
Depth (mm)
500
400
300
200
100
0
−100
−200
−300
−400
−500
−150
−100
−50
0
50
100
150
Angle (deg)
(a)
5000
4000
3000
2000
1000
0
0
20
40
60
80
100
120
140
160
180
(b)

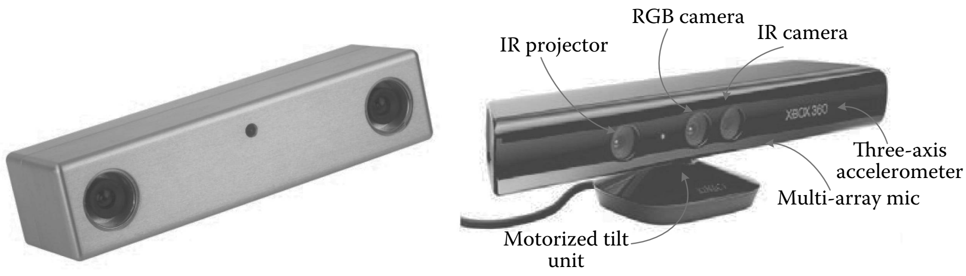
FIGURE 13.7
Typical laser distance finders: (a) the Hokuyo URG-04LX sensor and (b) SICK LMS 200 2-D scanner.

can be thought of as a miniature sonar that uses light instead of sound to create 2-D maps of the neighborhood of objects close to the robot. The scanner has an angular scanning range of 240°, and the angular resolution is approximately 0.36° with a scanning refresh rate of up to 10 Hz. Distances ranging from 20 mm to 4 m may be measured. The SICK LMS 200 2-D scanner has a horizontal range of 180° with a maximum resolution of 0.5°.

Another way of measuring distance is to utilize multiple cameras. This approach is termed stereo vision. Figure 13.8 shows two commercial stereo vision sensors: BumbleBee® 2 and Microsoft Kinect.

An important benefit of a stereo camera is that it can provide object identification and distance data simultaneously. A stereo camera has two sets of cameras as shown in Figure 13.9. Suppose that a point *P* is observed by two cameras simultaneously. Its projections to the image planes of the two cameras are at p and p', as shown. Note that O and O' are the optical centers of these two image planes. The line l', which is in image II, is called the epipolar line associated with point p of image I. Also, eL and eR are called epipoles of the two cameras. The epipolar eR is the projection of the optical center OL in the right camera frame.

Epipolar constraint [20] states that if p and p' are the projections of the same point P in different cameras, then p' must lie on the epipolar line associated with p. This result plays a fundamental role in stereo vision. The epipolar geometry can be simplified if the two


RGB camera
IR camera
IR projector
XBOX 360
Three-axis accelerometer
Multi-array mic
Motorized tilt unit

(a) (b)

FIGURE 13.8
Stereo vision sensors: (a) BumbleBee®2 stereo camera and (b) Microsoft Kinect.

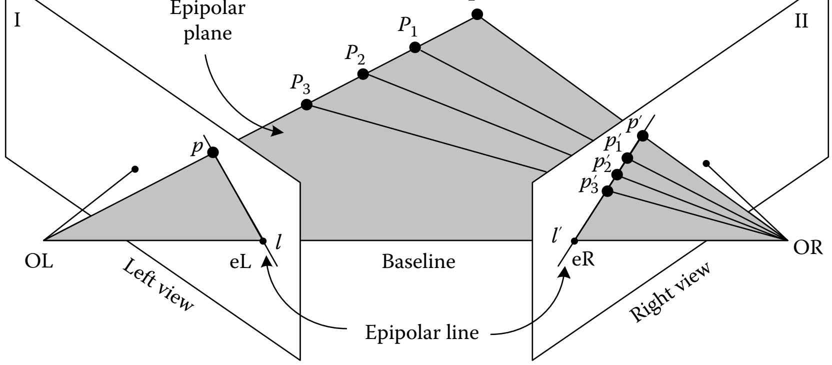

I
Epipolar plane
P
P_1
P_2
P_3
II
p
p'
p_1'
p_2'
p_3'
l
l'
OL
eL
Baseline
eR
OR
Left view
Right view
Epipolar line

FIGURE 13.9
Model of a stereo camera.

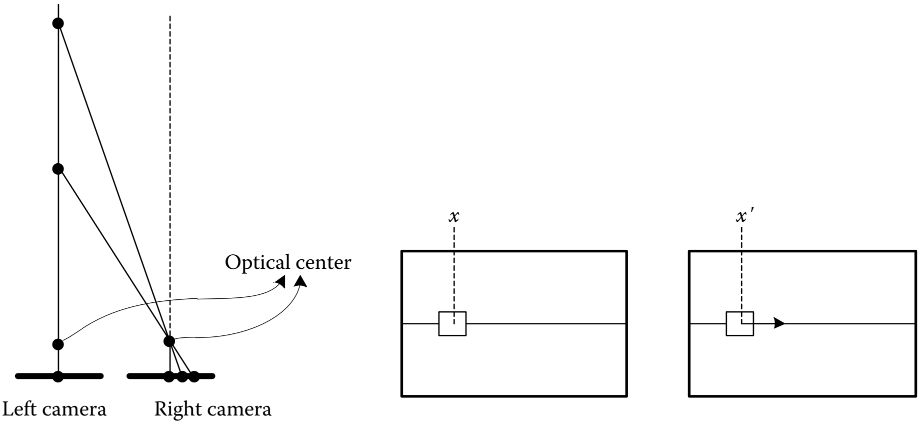


FIGURE 13.10
Simplified case of epipolar geometry.

camera image planes coincide as shown in Figure 13.10. Then, the epipolar lines also coincide. Furthermore, the epipolar lines are parallel to the line between the focal points and, in practice, can be aligned with the horizontal axes of the two images. This means that for each point in one image, its corresponding point in the other image can be found by looking only along a horizontal line.

The distance can be calculated by measuring the disparity, which is defined as

$$d = u_L - u_R = f\frac{B}{Z} \rightarrow Z = f\frac{B}{d} \tag{13.1}$$

Here, disparity = k/depth, and k is a system parameter, which can be acquired by calibration. By introducing the epipolar geometry and its constraints, the stereo vision problem is converted into a feature identification problem. In other words, the distance between the camera and the object in terms of the camera coordinates can be found by finding the position difference of the object in the right and left images (Figure 13.10). Some applications of stereo vision are given in [21–23].

13.3 Visual Servoing

13.3.1 Basic Categories of Visual Servoing

In general, visual servo control may be classified into the image-based approach and the position-based approach [1]. The two approaches share a common control block diagram with a slight difference in the control feedback loop and the reference input (Figure 13.3).

The control scheme entails minimization of the error by controlling the motion of the robot. The error may be defined by

$$e(t) = s((t), c) - s_{\mathrm{r}} \tag{13.2}$$

where $e(t)$ = error; s = measurement or pose estimation from the image; $i(t)$ = image data; c = camera model; and s_r = reference input. Once $s(i(t), c)$ and s_r are decided, a velocity controller can be designed as

$$\dot{s} = L\xi_c^c \tag{13.3}$$

where $\dot{s}$ is the velocity of the feature (feature points in IBVS and pose of the camera coordinate frame in PBVS); L is the interaction matrix; and ξ_c^c represents the angular and linear velocities of the camera with respect to the camera coordinate frame. Equations 13.2 and 13.3 may be combined and written as

$$\dot{e}(t) = L\xi_c^c \tag{13.4}$$

where $\dot{e}(t)$ = time variation of the error. The control law can be derived by using the square-of-error norm as a candidate Lyapunov function:

$$\dot{s} = L^{-1}(-ke) \tag{13.5}$$

13.3.2 Modeling of Visual Servo System

Modeling of a mobile manipulation system usually consists of the kinematic modeling of robots and the modeling of cameras. The former incorporates extensions of the principles of rigid body motions and homogeneous transformation [24], and the latter incorporates modeling of perspective projection of a pinhole camera [25].

13.3.2.1 Kinematic Modeling of Robots

Traditionally, the rigid body motions of a robot can be represented by the matrix

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, \; R \in SO(3), d \in \mathbb{R}^3 \tag{13.6}$$

where R represents a 3 × 3 rotational matrix, and d is a distance vector. The matrix in Equation 13.6 is called a homogeneous transformation, which represents the rotational and displacement relationships between two coordinate frames. It is a 4 × 4 matrix. The last row of the matrix consists of three zeros and a one, and remaining elements are composed of a rotation matrix and a position vector.

In robotic applications, a commonly used convention for defining the coordinate frames of reference and generating the homogeneous transformation matrices is the Denavit-Hartenberg (DH) convention. It was introduced by Denavit and Hartenberg [7] to simplify the modeling procedure in generating forward kinematics (i.e., expressing the end effector movement in terms of the joint movements) of a robot. There are four parameters in this representation: θ_i, d_i, a_i, and α_i. The detailed definitions of these four parameters are given in Table 13.1.

TABLE 13.1

Denavit-Hartenberg Convention

Parameter	Axis	Description
θ_i	z_{i-1}	Joint angle: the angle from x_{i-1} to x_i measured about z_{i-1}. Variable for revolute joints.
d_i	z_{i-1}	Link offset: distance along z_{i-1} from o_{i-1} to the intersection of the axes x_i and z_{i-1}. Variable for prismatic joints.
a_i	x_i	Link length: distance along x_i from the intersection of the axes x_i and z_{i-1} axes to o_i. Constant perpendicular distance between z_{i-1} and z_{i-1}.
α_i	x_i	Link twist: the angle from z_{i-1} to z_i measured about x_i. Constant angle between z_{i-1} and z_i.

Suppose that the coordinates are assigned based on the two DH rules:

(DH1): The axis x_i is perpendicular to the axis z_{i-1} ($x_i \perp z_{i-1}$).
(DH2): The axis x_i intersects the axis z_{i-1} ($x_i \cap z_{i-1} \neq \varnothing$).

Then, there exists a unique homogeneous transformation matrix that takes the coordinates from one frame to the base frame, as given by:

$$H = \mathrm{Rot}_{z,\theta}\mathrm{Trans}_{z,d}\mathrm{Trans}_{x,a}\mathrm{Rot}_{x,\alpha} =$$

$$\begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13.7}$$

where

$\mathrm{Rot}_{z,\theta}$ = rotational matrix about the z axis
$\mathrm{Trans}_{z,d}$ = translational matrix along the z axis
$\mathrm{Trans}_{x,a}$ = translational matrix along the x axis
$\mathrm{Rot}_{x,\alpha}$ = rotational matrix

Figure 13.11 shows an example of coordinate assignment according to the DH rules where $x_i \perp z_{i-1}$ and $x_i \cap z_{i-1} \neq 0$.

If the coordinates are defined according to the DH convention, the forward kinematic model of the robot can be represented in the form given in Equation 13.7.

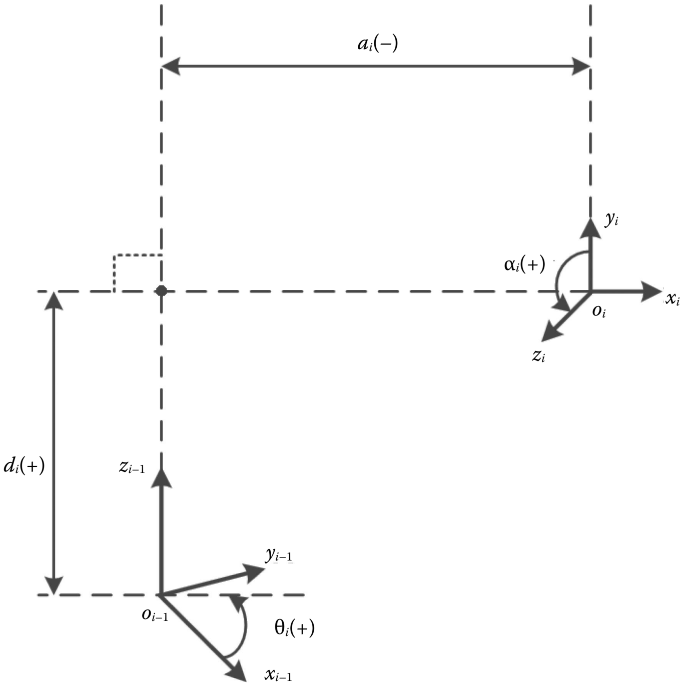


FIGURE 13.11
An example of coordinate frames satisfying DH convention.

13.3.2.2 Camera Modeling

The simplest model of a camera is the pinhole camera model, which is shown in Figure 13.12a. In this model, light from a point on the object (P) in the work space passes through a pinhole and projects onto the camera image surface called the image plane. For an ideal pinhole camera, the camera plane is located at a distance f (focal length) behind the pinhole. The origin of the coordinate frame is called the center of projection in the image plane. The intersection of the z-axis and the image plane is known as the principal point.

Figure 13.12b shows a pinhole camera model with the reflected image plane model [18]. This model has the advantage that the projected object (x') in the image plane is reversed when compared with the object in the world space (X). Therefore, the modified pinhole camera model simply shifts the image plane to the front of the principal point by $2f$. In this model, the image reversion problem is resolved. Meanwhile, the size of the projected object does not change in view of the property of similar triangles. If the camera focal length f is known, and the coordinates of the point object in terms of the camera frame are known as well, which is denoted by (x_c, y_c, z_c), we can determine the coordinates of the point in the real world in terms of the camera frame by sensing the position of the projected point in the image plane (u, v):

$$k\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} u \\ v \\ f \end{bmatrix} \tag{13.8}$$

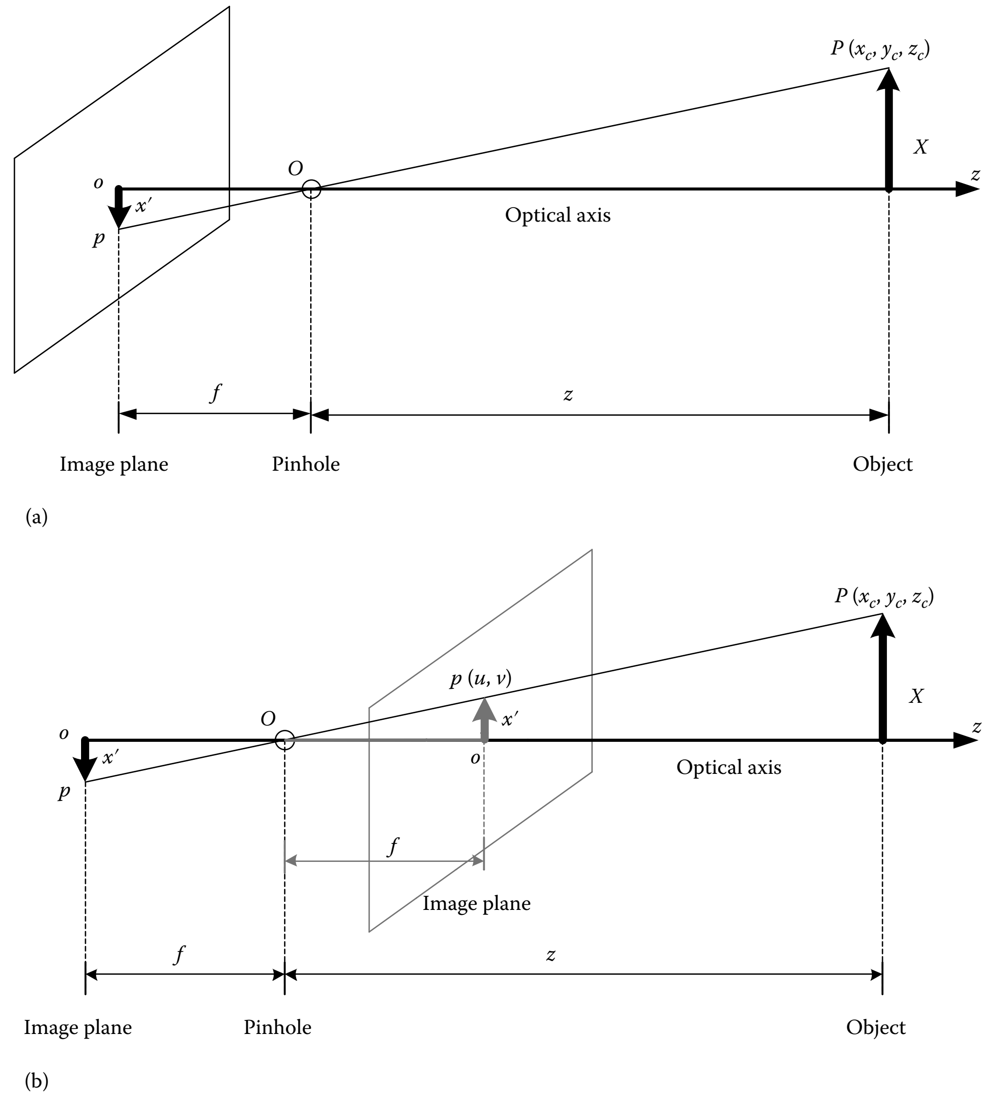

P (x_c, y_c, z_c)
X
z
O
o
x′
p
Optical axis
f
z
Image plane
Pinhole
Object
(a)
P (x_c, y_c, z_c)
p (u, v)
x′
X
z
O
o
x′
p
o
Optical axis
f
Image plane
f
z
Image plane
Pinhole
Object
(b)

FIGURE 13.12
(a) Pinhole camera model and (b) pinhole camera model with reflected image plane.

where u,v are the coordinates of the projected point in the image plane. It is seen that these equations can help identify the 3-D position of a point in the camera frame if u and v are measureable. However, it is not easy to measure these variables. The sensor of a digital camera is a 2-D array whose elements are called pixels as shown in Figure 13.13. These pixels are not continuous, and there exist gaps among the pixels. Moreover, a pixel element is usually rectangular and not square, especially in low-cost cameras. The dimensions of the pixel element are described by s_x and s_y in Figure 13.13. Also, the origin of the image coordinate frame, which is the principal point, is usually not the center of the image/pixel plane. There exist offsets in real situations.

In Figure 13.12b, P is a point in the work environment with coordinates $(x, y, z)^c$ relative to the camera frame, and p is the projection of P on the image plane with coordinates (u, v)

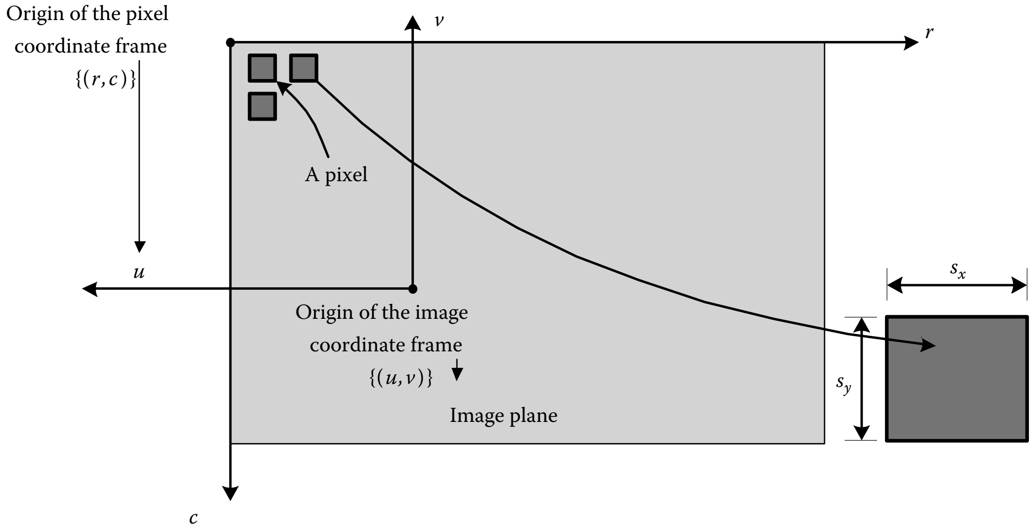

Origin of the pixel coordinate frame
$\{(r,c)\}$
v
r
A pixel
u
Origin of the image coordinate frame
$\{(u,v)\}$
Image plane
s_x
s_y
c

FIGURE 13.13
Image plane and pixel plane.

relative to the image plane frame and with coordinates (r, c) relative to the pixel coordinate frame. The distance between the origin of the camera frame and the image plane is denoted by λ, and the coordinates of the principal point are (o_r, o_c) with respect to the pixel coordinate frame. The coordinate transformation between the frames is given by

$$\begin{cases} u = -s_x(r - o_r) \\ v = -s_y(c - o_c) \end{cases} \tag{13.9}$$

$$\begin{cases} r = -f_x \dfrac{x^c}{z^c} + o_r \\ c = -f_y \dfrac{y^c}{z^c} + o_c \end{cases} \tag{13.10}$$

where, s_x and s_y are the horizontal and vertical dimensions, respectively, of a pixel as given by $f_x = \dfrac{\lambda}{s_x}$ and $f_y = \dfrac{\lambda}{s_y}$.

Let p be a feature point on the image plane with coordinates (u, v). The moving velocity of p can be expressed in terms of the camera velocity using the interaction matrix L [7] as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = L\xi_c^c = \begin{bmatrix} -\dfrac{\lambda}{z^c} & 0 & \dfrac{u}{z^c} & \dfrac{uv}{\lambda} & -\dfrac{\lambda^2 + u^2}{\lambda} & v \\ 0 & -\dfrac{\lambda}{z^c} & \dfrac{v}{z^c} & \dfrac{\lambda^2 + v^2}{\lambda} & -\dfrac{uv}{\lambda} & -u \end{bmatrix} \xi_c^c \tag{13.11}$$

In Equations 13.9 through 13.11, there are several unknowns: the dimensions of the sensors (s_x and s_y), focal length of the camera (f_x and f_y), the location of the principal points (o_r and o_c), and the velocity of the camera with respect to the camera frame $\left(\xi_c^c\right)$. These parameters can be determined through camera calibration, which involves finding two groups of parameters: (a) intrinsic camera parameters and (b) extrinsic camera parameters.

13.3.2.3 Camera Parameters

Because cameras are usually rigidly attached to the robots or mounted on tripods, the rotational matrix R_c^w and the translational vector T_c^w can be easily determined. Therefore, if we know the 3-D position of a point P in the camera frame, the coordinates of this point in the world frame can be determined by using

$$P^w = R_c^w P^c + T_c^w \tag{13.12}$$

or, conversely,

$$P^c = R_w^c \left(P^w - T_c^w\right) = R_w^c P^w - R_w^c T_c^w \tag{13.13}$$

where R_w^c and $R_w^c T_c^w$ are called the extrinsic camera parameters. We will denote them by R and T, respectively, for brevity. Consider the rotational matrix:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{13.14}$$

and the translational vector:

$$T = [T_x \; T_Y \; T_z]^T \tag{13.15}$$

with respect to the world coordinate frame. The coordinates of the point in terms of the camera frame can be found by

$$x^c = r_{11}x^w + r_{12}y^w + r_{13}z^w + T_x \tag{13.16}$$

$$y^c = r_{21}x^w + r_{22}y^w + r_{23}z^w + T_y \tag{13.17}$$

$$z^c = r_{31}x^w + r_{32}y^w + r_{33}z^w + T_z \tag{13.18}$$

By substituting Equations 13.16 and 13.17 into Equation 13.18, we obtain

$$r - o_r = -f_x \frac{x^c}{z^c} = -f_x \frac{r_{11}x^w + r_{12}y^w + r_{13}z^w + T_x}{r_{31}x^w + r_{32}y^w + r_{33}z^w + T_z} \tag{13.19}$$

$$c - o_c = -f_y \frac{x^c}{z^c} = -f_y \frac{r_{21}x^w + r_{22}y^w + r_{23}z^w + T_y}{r_{31}x^w + r_{32}y^w + r_{33}z^w + T_z} \tag{13.20}$$

By combining Equations 13.18 and 13.19, we obtain

$$f_x(c - o_c)(r_{11}x^w + r_{12}y^w + r_{13}z^w + T_x) = f_y(r - o_r)(r_{21}x^w + r_{22}y^w + r_{23}z^w + T_y) \tag{13.21}$$

This final relationship has nine unknowns ($f_x, f_y, r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, T_x, T_y$). Here R and T are the extrinsic parameters, which denote the coordinate transformations from the 3-D world coordinates to the 3-D camera coordinates. Equivalently, the extrinsic parameters define the position of the camera center and the camera heading in world coordinates, which may be determined by the homogeneous transformation as discussed previously. The intrinsic parameters are determined by a camera calibration tool as presented in [26].

13.4 Case Study of Visual Servoing

Figure 13.14 illustrates an object-grasping task using visual servoing by a mobile robot [27]. The purpose of the mobile robot in the work space, which is shown in Figure 13.14, is to find the object of interest and grasp it using the onboard manipulator.

It contains a mobile base, a manipulator, and different types of sensors (web camera, laser, sonar, and stereo camera). The web camera and the stereo cameras are in the eye-in-hand configuration. The distance is measured by using a laser distance finder, sonar, and stereo camera instead of using estimation.

Both simulation and physical experimentation are carried out in a laboratory environment to evaluate the performance of the methodologies and the robotic system.

13.4.1 System Modeling

The goal of image-based visual servoing (IBVS) is to eliminate the position errors of the feature points in the image plane by controlling the velocities of the robot joints. To this end, the inputs and the outputs of the models are taken as the linear/angular velocities of the robot joints and the positions of the feature points in the image plane, respectively.

Four sets of coordinate frames are defined now, namely the robot frame, the camera frame, the image plane frame, and the pixel coordinate frame, in order to derive the IBVS control law for the mobile robot. The relationship between the first two coordinate frames is shown in Figure 13.15.

For modeling the robot, first, the relationship between the velocity of the end effector with respect to the robot base frame and the velocity of each joint is derived as given by

$$\xi_t^r = J_1 \dot{q} \tag{13.22}$$

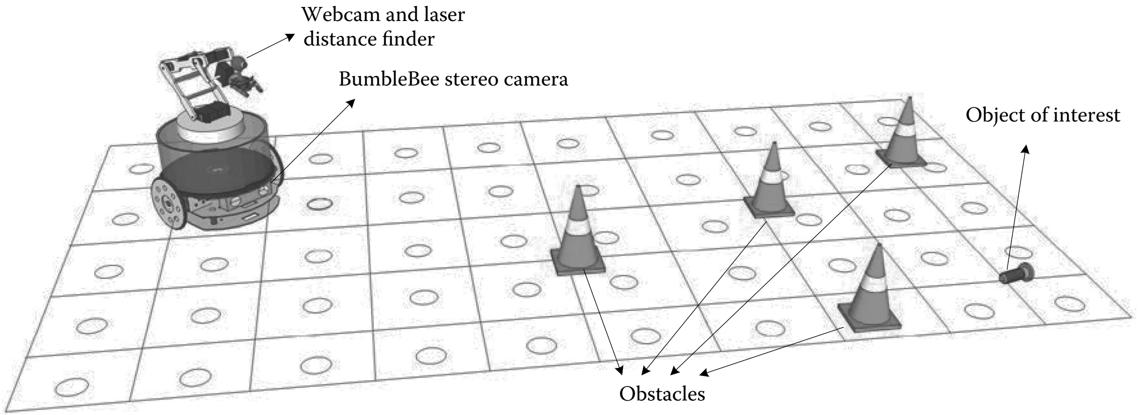

Webcam and laser distance finder
BumbleBee stereo camera
Object of interest
Obstacles

FIGURE 13.14
An example of object grasping by a mobile robot.

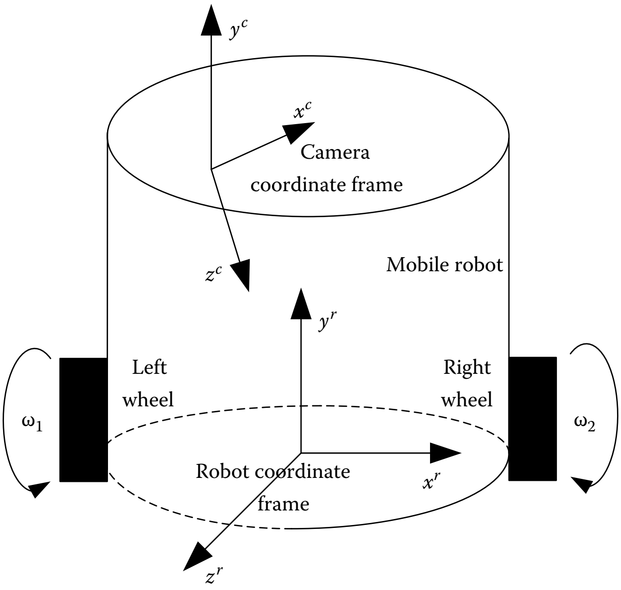

y^c
x^c
Camera
coordinate frame
z^c
Mobile robot
y^r
Left
wheel
Right
wheel
ω_1
ω_2
Robot coordinate
frame
x^r
z^r

FIGURE 13.15
Robot frame and camera frame.

where J_1 is the Jacobian matrix [1], and $\dot{q}$ is a vector that represents the linear/angular velocities of the joints of the robot. The velocity of the end effector with respect to its own frame is given by

$$\xi_t^t = G^{-1}\xi_t^r \tag{13.23}$$

where $= \begin{bmatrix} R_t^r & 0_{3\times3} \\ 0_{3\times3} & R_t^r \end{bmatrix}$. Because the camera is rigidly attached to the end effector, the camera frame and the end effector frame have a constant relationship of homogeneous transformation. Therefore, one can establish the relationship between the camera velocity and the end effector velocity as

$$\xi_t^t = J_2\xi_c^c \tag{13.24}$$

where $J_2 = \begin{bmatrix} R_c^t & s\left(d_c^t\right)R_c^t \\ 0_{3\times3} & R_c^t \end{bmatrix}$. Next, the relationship between the camera velocity and the velocities of the feature points in the image plane is written by using the interaction matrix as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = L\xi_c^c \tag{13.25}$$

Then, the relationship between the velocities of the feature points in the image plane and the velocities of the robot joints can be found through Equations 13.22 through 13.25 as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = LJ_2^{-1}G^{-1}J_1\dot{q} \tag{13.26}$$

Finally, the relationship between the positions of the feature points in the image plane and velocities of the joints can be derived through the integration of Equation 13.26, which is the mathematic model for the image-based visual servo robotic system. Because the system has a mobile base, the model can be simplified as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = LG^{-1}J\dot{q} \tag{13.27}$$

Because the camera is rigidly attached to the mobile base, the extrinsic parameters are fixed. Hence, Equation 13.27 can be rewritten by substituting the intrinsic and extrinsic parameters and transforming into the Laplace domain:

$$\begin{bmatrix} \bar{U}(s) \\ \bar{V}(s) \end{bmatrix} = \begin{bmatrix} \dfrac{M_{11}(t)}{s} & \dfrac{M_{12}(t)}{s} \\ \dfrac{M_{21}(t)}{s} & \dfrac{M_{22}(t)}{s} \end{bmatrix} \begin{bmatrix} V_l(s) \\ \Omega(s) \end{bmatrix} \tag{13.28}$$

where M_{11}, M_{12}, M_{21}, M_{22} become constant values after substitution and linearization.

13.4.2 Traditional Image-Based Visual Servoing

Now a traditional image-based eye-in-hand visual-servo control law is developed for a mobile robot. The underlying concept of this IBVS is that the controller will continuously adjust the speed of the robot so that the coordinates (u, v) of the feature point are moved toward the desired position (u_d, v_d) on the image. In particular, the error vector of the image feature point is given by

$$e = \begin{bmatrix} u - u_d \\ v - v_d \end{bmatrix} = \begin{bmatrix} -s_x(r - r_d) \\ -s_y(c - c_d) \end{bmatrix} \tag{13.29}$$

where s_x and s_y are the dimensions of the pixel of the image sensor, which is a charge-coupled device (CCD) in this case.

The velocity of the feature point can be expressed as

$$\dot{e} = \begin{bmatrix} \dfrac{\mathrm{d}(u - u_d)}{\mathrm{d}t} \\ \dfrac{\mathrm{d}(v - v_d)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \tag{13.30}$$

By substituting Equations 13.27 and 13.29 into Equation 13.30, the following two equations are obtained:

$$\dot{e} = LG^{-1}J\dot{q} = M_2\,[v\ \omega] \tag{13.31}$$

In view of Equation 13.31, assuming that the error dynamics obeys $\dot{e} = -ke$, a proportional controller based on the Lyapunov method is designed according to

$$[v\ \omega] = M^{-1}(-ke) \tag{13.32}$$

Here, k is the scalar proportional gain, with $k > 0$. The control law for the mobile robot is obtained by substituting Equation 13.29 into Equation 13.32, as

$$\Omega = -kM^{-1}\begin{bmatrix} -s_x(r - r_d) \\ -s_y(c - c_d) \end{bmatrix} \tag{13.33}$$

where Ω = velocity vector of the mobile robot. In Equation 13.32, the pixel coordinates r and c can be measured directly from the current image using the available image-processing software. Therefore, the desired velocities can be directly computed from the image measurements. Moreover, it is noted that the developed controller guarantees asymptotic stability of the closed-loop system in view of the error equation $\dot{e} = -ke$.

A vision-based mobile grasping system as developed in our laboratory is employed to validate the visual-servo controller. In the experiment, the mobile robot employs its onboard CCD camera to continuously observe the position of the target object (toothpaste tube with a red cap) shown in Figure 13.15, with the ACTS color blob–tracking software, and computes the visual error e on the image plane. Next, the IBVS controller determines the desired wheel speeds ω_1 and ω_2 using Equation 13.32 and accordingly commands the low-level controller of the mobile robot to gradually move the robot close to the target object. When the robot is sufficiently close to the object, it grasps the object with the assistance of the onboard 5DOF camera. The overall process of the experiment is shown in Figure 13.16.

The trajectory of the visual feature point on the image plane is shown in Figure 13.17.

In Figure 13.17, the initial position of the visual feature is close to the top right corner of the image and then moves directly toward the desired position at the bottom left corner.

The position and heading history of the robot in the entire process is shown in Figure 13.18, the pixel errors on the image plane are shown in Figure 13.19, and the control inputs are given in Figure 13.20.

From Figures 13.16 through 13.20, it is clear that the developed IBVS controller is able to effectively drive a wheeled mobile robot toward a target object until its CCD camera observes the desired visual features.

13.4.3 Adaptive Nonlinear Model Predictive Control

Over the years, nonlinear model predictive control (NMPC) has been popular in the field of control, particularly as a control method for robots. The most important advantage of NMPC is that it can accommodate various input/output constraints in a nonlinear plant

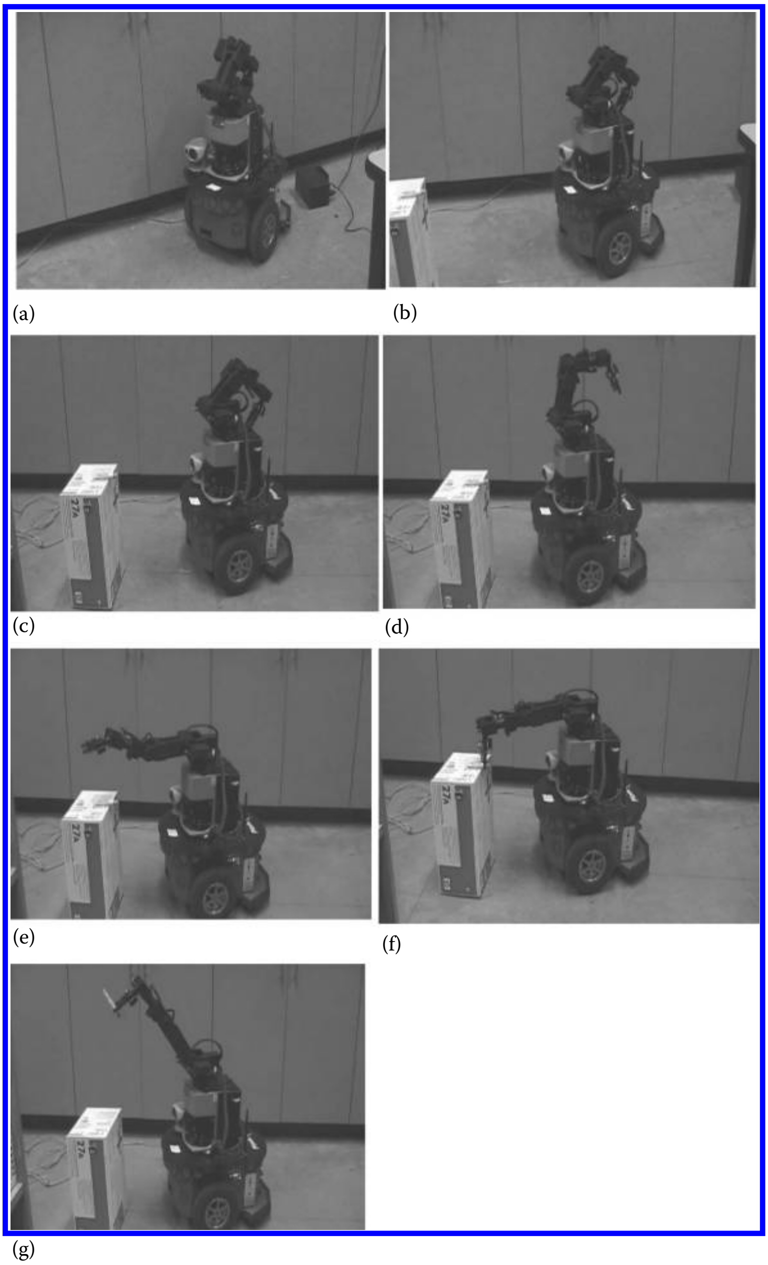


FIGURE 13.16
Vision-based autonomous grasping using a wheeled mobile robot and the developed IBVS controller.

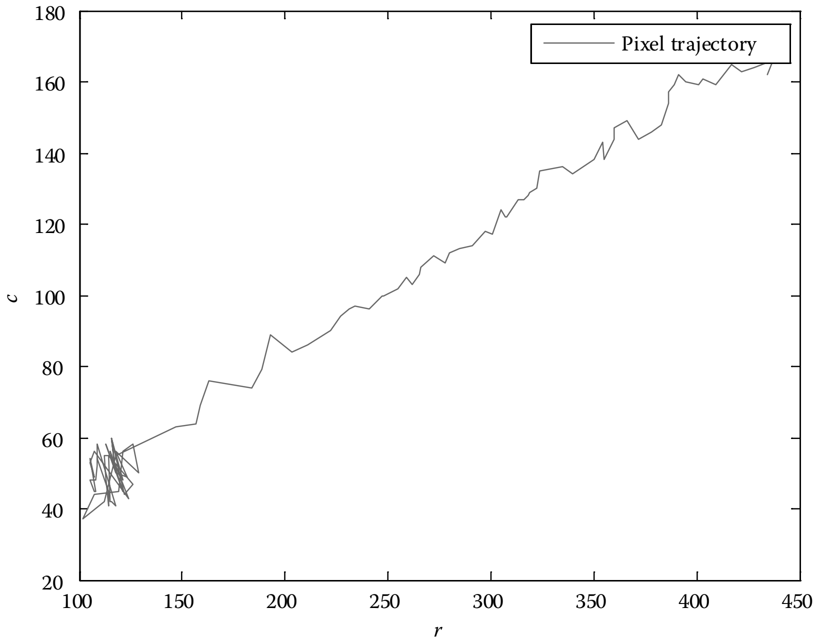


FIGURE 13.17
Trajectory of the visual feature point (object) on the image plane when the robot approached the object and attempted to grasp it.

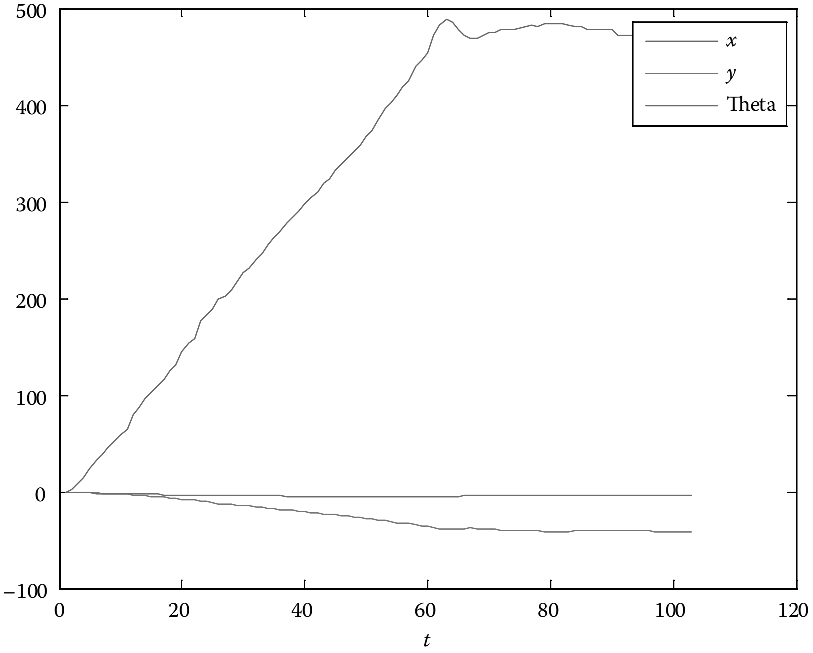


FIGURE 13.18
Trajectory (position and heading) of the mobile robot in the physical environment when it carried out the mobile manipulation task.

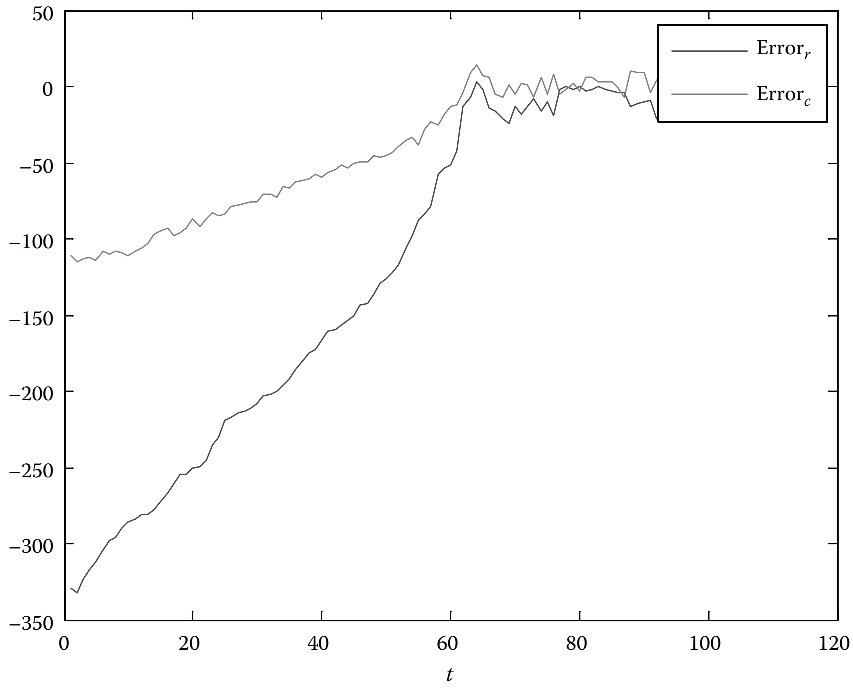


FIGURE 13.19
Visual errors on the image plane when the robot approached the object and attempted to grasp it.

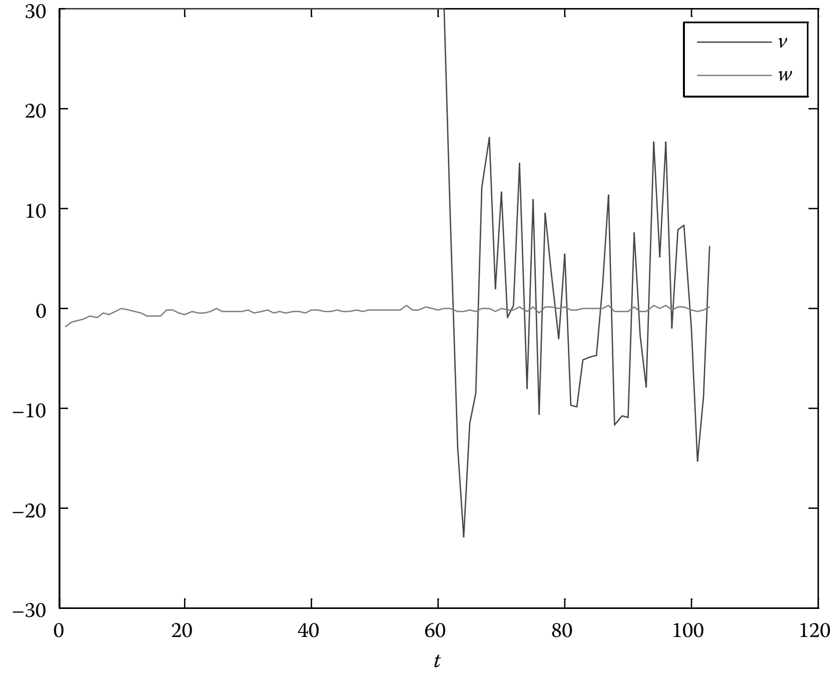


FIGURE 13.20
The control inputs of the plant when the new visual servo controller is working.

when calculating the optimal control inputs. The basic idea of NMPC is to calculate an optimal control input sequence for the plant through model prediction so as to minimize the cost function and meet various constraints on the plant. More details of NMPC can be found in [28].

As discussed in the previous section, the mobile visual servo implementation that is shown in Figure 13.14 is related to multiple constraints and has a nonlinear model. So it is quite natural to apply NMPC to a task of visual servoing. However, the model described by Equation 13.31 is highly nonlinear and time varying, and the classical NPMC algorithm requires a linear model to predict the behavior of a nonlinear plant [28]. A popular solution for this problem is to find a constant linear model at the equilibrium point (operating point) of the plant and use this constant model in the NMPC controller. Although this approach works well in visual servoing of fixed-base robotic arms, it will fail in visual servoing of mobile robots. This is because mobile robots usually move over large distances with varying arm configurations, which can place the current operating point far away from the original equilibrium point. Therefore, a more advanced NMPC scheme is needed for vision-based mobile manipulation.

In this section, a new NMPC scheme, called adaptive nonlinear model predictive control (A-NMPC), is introduced to meet the above challenge. The A-NMPC control approach is presented in Figure 13.21.

In Figure 13.21, the A-NMPC controller includes four units: the model updating unit, linearized model, constraint unit, and optimizer unit. In particular, the current position of the target object in the image (i.e., $[u, v]^T$) and the current depth z are continuously measured by the CCD camera and the laser distance finder and are sent into the model updating unit. Then the linearized model of the plant is updated by the model updating

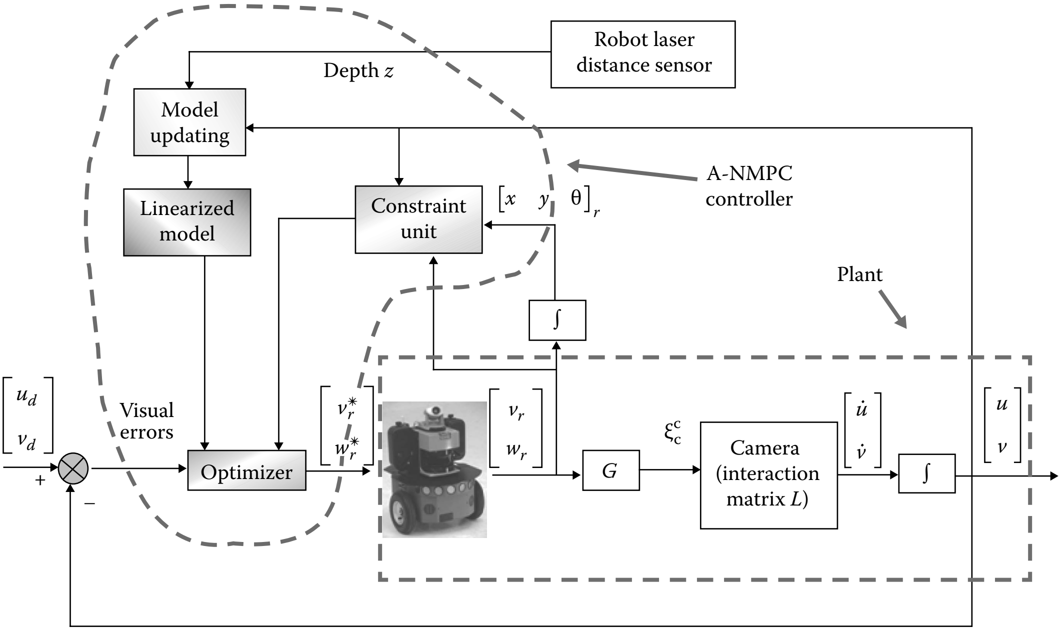

Robot laser distance sensor
Depth z
Model updating
Linearized model
Constraint unit
$[x \quad y \quad \theta]_r$
A-NMPC controller
Plant
$\int$
$\begin{bmatrix} u_d \\ v_d \end{bmatrix}$
+
−
Visual errors
Optimizer
$\begin{bmatrix} v_r^* \\ w_r^* \end{bmatrix}$
$\begin{bmatrix} v_r \\ w_r \end{bmatrix}$
G
ξ_c^c
Camera (interaction matrix L)
$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}$
$\int$
$\begin{bmatrix} u \\ v \end{bmatrix}$

FIGURE 13.21
Scheme of adaptive nonlinear model predictive control (A-NMPC) for vision-based mobile manipulation.

unit with the latest u, v, and z so that the model always tracks the nonlinear plant. In addition, three kinds of constraints are considered and set up by the constraint unit, which will constrain $[v, \omega]^T$ (the translational and rotational velocities of the robot), $[x, y, \theta]^T$ (the location and heading of the mobile robot in the environment), and $[u, v]^T$ (the position of the target object in the image). Finally, based on the latest linearized model, the constraint requirements and the current outputs of the plant, the optimizer unit will calculate an optimal control input sequence using the quadratic programming algorithm (QP) that is described in [28].

The cost function of the A-NMPC controller is given by

$$C(k) = \sum_{i=1}^{H_p} \left\| \begin{matrix} \hat{u}(k+i\,|\,k) - u_d \\ \hat{v}(k+i\,|\,k) - v_d \end{matrix} \right\|_{Q(i)}^{2} + \sum_{i=0}^{H_{u-1}} \left\| \begin{matrix} \Delta\hat{v}_l(k+i\,|\,k) \\ \Delta\hat{\omega}(k+i\,|\,k) \end{matrix} \right\|_{R(i)}^{2} \tag{13.34}$$

where H_p is the prediction horizon, H_u is the control horizon, (u_d, v_d) is the desired position of the target object in the image, and $Q(i)$ and $R(i)$ are the weighting matrices with $Q(i) \geq 0$ and $R(i) \geq 0$.

The A-NMPC controller, as shown in Figure 13.21, considers three kinds of constraints as follows:

$$u_{\min} \leq u(t) \leq u_{\max} \quad \text{and} \quad v_{\min} \leq v(t) \leq v_{\max} \tag{13.35}$$

(visibility constraints)

$$x(t) \leq x_{\max}, \; y(t) \leq y_{\max}, \; \theta_{\min} \leq \theta(t) \leq \theta_{\max} \tag{13.36}$$

(robot location constraints)

$$v_l(t) \leq v_{l\max} \quad \text{and} \quad \omega_l(t) \leq \omega_{l\max} \tag{13.37}$$

(robot velocity constraints)

The mobile robot has an onboard computer inside its case, which provides sufficient computational resources for running the A-NMPC controller in real time. In addition, the ACTSTM color blob–tracking software is used to track the position of the target object in the image in real time. The NMPC controller is first designed and tested in the MATLAB® MPC Toolbox™. Then it is compiled into a DLL file using the MATLAB Compiler™, which can be further loaded and called by the C++ main program to control the motion of the robot. The C++ main program also calls the ACTSTM software and the mobile robot APIs to get the current position, velocity, and depth information, which are employed to update the linearized model online.

In the first experiment, an A-NMPC controller without considering any constraints is tested. The controller parameters are listed as follows: Sampling time t_s = 100 ms, H_p = 10, H_u = 3, $Q(i) = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$, and $R(i) = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$. The experimental results are presented in Figure 13.22.

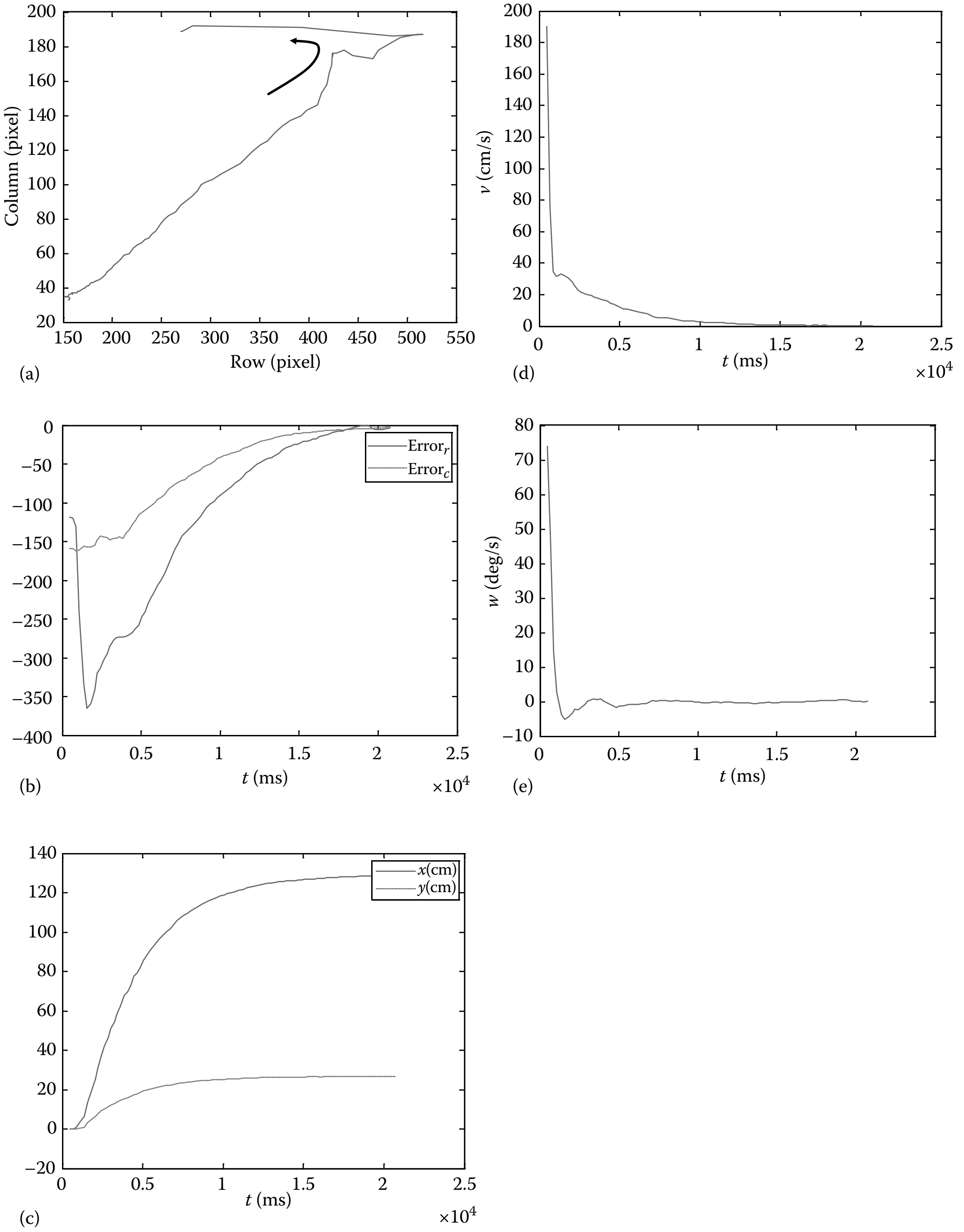
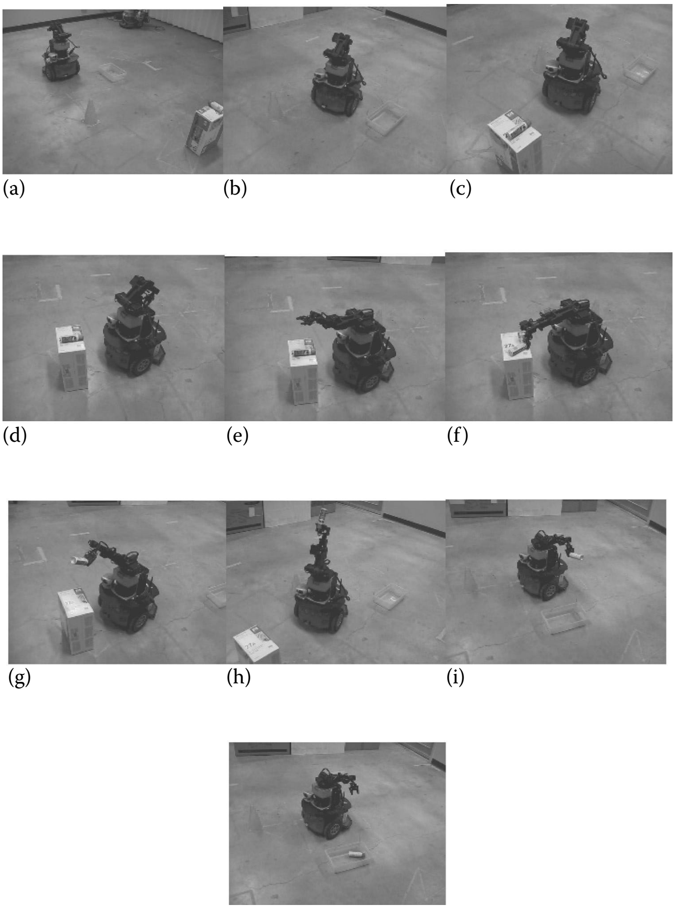


FIGURE 13.22
Experimental results of a mobile visual servo system using unconstrained A-NMPC. (a) Trajectory of the target object in the image. (b) Pixel errors on the image plane. (c) History of the mobile robot location. (d) History of the robot translational velocity (control input). (e) History of the robot rotational velocity (control input).

In the experiment, the A-NMPC has the following constraint parameters: $20 \leq r \leq 450$, $20 \leq c \leq 300$, $-20\ \text{cm/s} \leq v_l \leq 20\ \text{cm/s}$, $-6\ \text{cm/s} \leq \omega \leq 6\ \text{cm/s}$, $y \geq -15$ cm. A series of still images grabbed from the experiment video are presented in Figure 13.23.

In Figure 13.23, a mobile robot with an onboard arm and a CCD camera is used to grasp a target object (a toothpaste tube with a red cap) in an environment with two obstacles. It is shown that the vision-based mobile manipulation system, commanded by the constrained A-NMPC visual servo controller developed in the present work, avoids the obstacles, moves toward the target object, and grasps it successfully. The system outputs and control inputs in this process of visual servoing are presented in Figure 13.24.

From Figure 13.24a, d, and e, it is clear that the system outputs $r(t)$ and $c(t)$ and the control inputs $v_l(t)$ and $\omega(t)$ are constrained within the desired limits, and Figure 13.24b shows that the errors quickly converge to zero. Therefore, from Figure 13.24, it can be concluded that the A-NMPC visual servo controller is quite successful in maintaining optimal control performance while satisfying various constraints (visibility constraints, velocity constraints, and so on) at the same time.


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j)

FIGURE 13.23
Vision-based autonomous grasping using a mobile robot and the constrained A-NMPC controller.

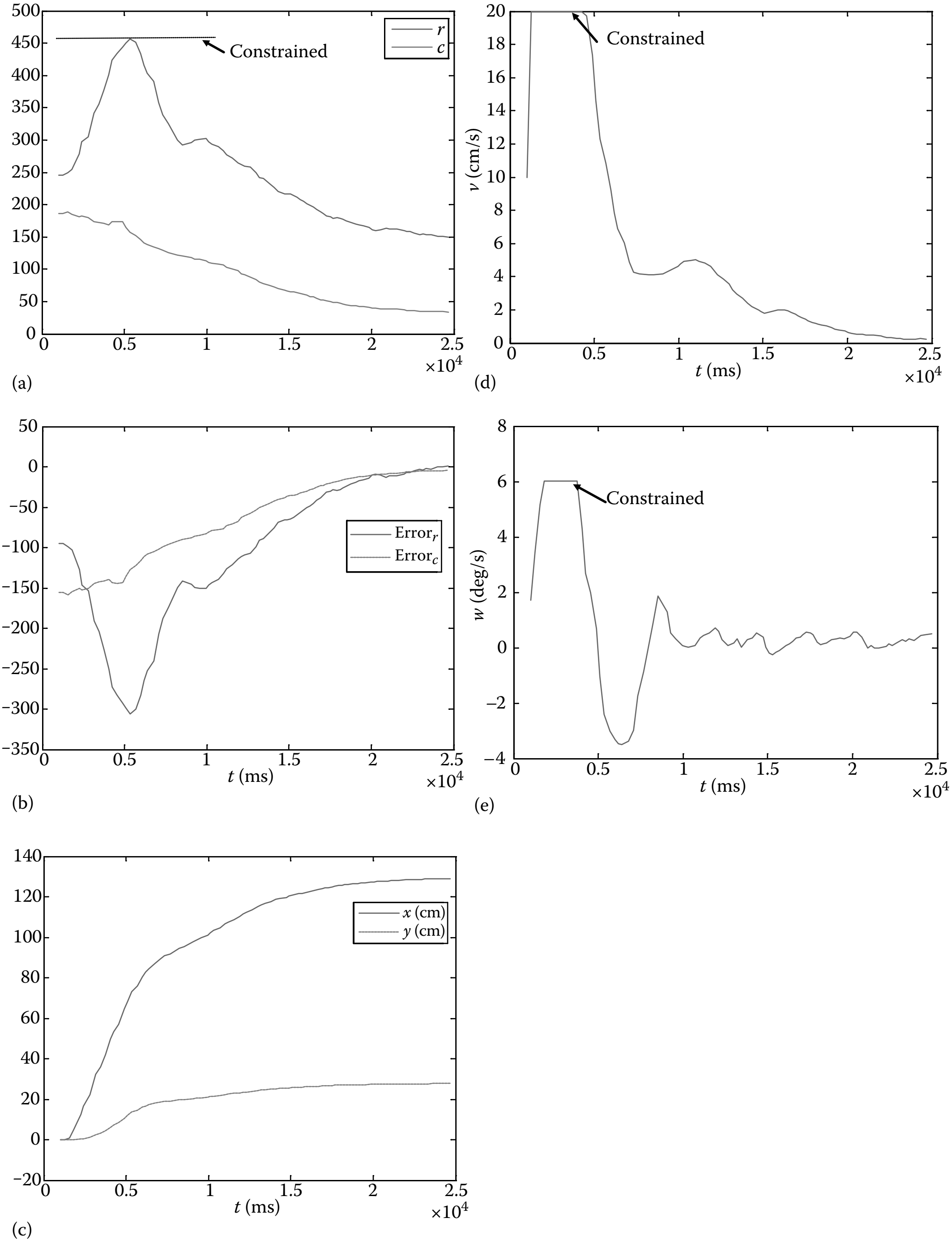


FIGURE 13.24
Mobile visual servoing using constrained A-NMPC. (a) System outputs (the pixel position of the target object in the image). (b) Pixel errors on the image plane. (c) History of the mobile robot location. (d) History of the robot translational velocity (control input). (e) History of the robot rotational velocity (control input).

13.5 Conclusions

Controlling the motion of a robot in order to carry out a specific task is a challenge in the application of autonomous mobile robotics. Visual servoing is a control solution for mobile robots that is able to work in an unknown, unstructured, and dynamic environment. In this chapter, the developments of research in visual servoing were outlined along with a fundamental formulation of the problem, which included mathematical modeling of robots and cameras and the controller design. Two case studies were presented to demonstrate that visual servoing was effective operating and handling autonomous tasks in a complex environment.

References

1. Chaumette, F. and Hutchinson, S., "Visual servo control part 1: Basic approaches," *IEEE Robotics and Automation Magazine*, Vol. 13, No. 4, pp. 82–90, 2006.
2. Agin, G. J., "Real time control of a robot with a mobile camera," *SRI International*, Technical Note 179, Menlo Park, CA, 1979.
3. Sanderson, A. C. and Weiss, L. E., "Image-based visual servo control using relational graph error signals," *Proceedings of IEEE*, pp. 1074–1077, 1980.
4. Hutchinson, S., Hager, G. D., and Corke, P. I., "A tutorial on visual servo control," *IEEE Transaction on Robotics and Automation*, Vol. 12, No. 5, pp. 651–670, 1996.
5. Patz, B. J., Papelis, Y., Pillat, R., Stein, G., and Harper, D., "A practical approach to robotic design for the DARPA Urban Challenge," *Journal of Field Robotics*, Vol. 25, No. 8, pp. 528–566, 2008.
6. Motemerlo, M., Becker, J., Shat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D. et al., "Junior: The Stanford entry in the urban challenge Montemerlo," *Journal of Field Robotics*, Vol. 25, No. 9, pp. 569–597, 2008.
7. Spong, M. W., Hutchinson, S., and Vidyasagar, M., *Robot Modeling and Control*, Wiley, Hoboken, NJ, 2005.
8. Hvilshj, M., Bgh, S., Nielsen, O., and Madsen, O., "Autonomous industrial mobile manipulation (AIMM): Past, present and future," *Industrial Robot*, Vol. 39, No. 2, pp. 120–135, 2012.
9. Joseph, M., Hsiao, K., Niemeyer, G., Chitta, S., and Kuchenbecker, K. J., "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, Vol. 27, No. 6, pp. 1067–1079, 2011.
10. Bohren, J., Rusu, R. B., Jones, E. G., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mösenlechner, L., Meeussen, W., and Holzer, S., "Towards autonomous robotic butlers: Lessons learned with the PR2," *Proceedings of IEEE International Conference on Robotics and Automation*, ICRA 2011, pp. 5568–5575, 2011.
11. Fuchs, M., Borst, C., Giordano, P. R., Baumann, A., Kraemer, E., Langwald, J., Gruber, R. et al., "Rollin' Justin—Design considerations and realization of a mobile platform for a humanoid upper body," *IEEE International Conference on Robotics and Automation*, ICRA 09, pp. 4131–4137, 2009.
12. Bluethmann, W., Ambrose, R., Diftler, M., Askew, S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., and Magruder, D., "Robonaut: A robot designed to work with humans in space," *Autonomous Robots*, Vol. 14, Nos. 2–3, pp. 179–197, 2003.
13. Uhlmann, J., Julier, S., Kamgar-Parsi, B., Lanzagorta, M., and Shyu, H., "NASA Mars rover: A testbed for evaluating applications of covariance intersection," *Proceedings of SPIE—The International Society for Optical Engineering*, Vol. 3693, pp. 140–149, 1999.

14. Kucera, J. and Sirak, M., “Bomb-clearing robots to enter full-rate production,” *Jane's Defence Weekly*, July 2005.
15. Kucera, J., “US Army seeks to hone bomb-clearing robots,” *Jane's Defence Weekly*, September 2005.
16. Bischoff, R., Huggenberger, U., and Prassler, E., “KUKA youBot—A mobile manipulator for research and education,” *IEEE International Conference on Robotics and Automation*, ICRA, 2011.
17. Vohr, F. and May, T., “Robot hardware design in the DESIRE project,” *Springer Tracts in Advanced Robotics*, Vol. 76, No. STAR, pp. 11–18, 2012.
18. Siegwart, R. and Nourbakhs, L. R., *Introduction to Autonomous Mobile Robots*, The MIT Press, Cambridge, MA; London, England, 2004.
19. De Silva, C. W., *Sensors and Actuators—Engineering System Instrumentation* (2nd Edition), Taylor & Francis, CRC Press, Boca Raton, FL, 2015.
20. Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer, New York, 2011.
21. Mutto, D. C., Pietro, Z., and Guido, C. M., “Fusion of geometry and color information for scene segmentation,” *IEEE Journal on Selected Topics in Signal Processing*, Vol. 6, No. 5, pp. 505–521, 2012.
22. Huntsberger, T., Aghazarian, H., Howard, A., and Trotz, D. C., “Stereo vision-based navigation for autonomous surface vessels,” *Journal of Field Robotics*, Vol. 28, No. 1, pp. 3–18, 2011.
23. Lee, C., Lim, Y., Kwon, S., and Lee, J., “Stereo vision-based vehicle detection using a road feature and disparity histogram,” *Optical Engineering*, Vol. 50, No. 2, pp. 1–23, 2011.
24. Crag, J. J., *Introduction to Robotics: Mechanics and Control* (3rd Edition), Prentice Hall, Upper Saddle River, NJ, 2004
25. Gonzalez, R. C. and Woods, R. E., *Digital Image Processing* (2nd Edition), Prentice Hall, Upper Saddle River, NJ, 2002.
26. Zhang, Z. Y., “Flexible camera calibration by viewing a plane from unknown orientations,” *The Proceeding of the Seventh IEEE International Conference on Computer Vision*, Vol. 1, pp. 666–673, 1999.
27. Lang, H., Khan, M. T., Tan, K. K., and de Silva, C. W., “Developments in visual servoing for mobile manipulation,” *Unmanned Systems*, Vol. 1, No. 1, pp. 143–162, 2013.
28. Camacho, E. F. and Bordons, C., *Model Predictive Control* (2nd Edition), Springer, London, UK, 2007.

14

Robotic Learning and Applications

Ying Wang and Bashan Zuo

CONTENTS

14.1 Introduction 530
14.2 Markov Decision Process (MDP) and *Q* Learning 530
14.3 Case Study: Multi-Robot Transportation Using Machine Learning 534
  14.3.1 Multi-Agent Infrastructure 534
  14.3.2 Cooperation Based on Machine Learning 536
    14.3.2.1 Reinforcement Learning 536
    14.3.2.2 Genetic Algorithms 538
    14.3.2.3 Integrating Reinforcement Learning with Genetic Algorithms 540
  14.3.3 Simulation Results 541
  14.3.4 Experimentation 543
14.4 Case Study: A Hybrid Visual Servo Controller Using *Q* Learning 545
  14.4.1 Vision-Based Mobile Robot Motion Control 546
    14.4.1.1 Control Scheme 546
    14.4.1.2 Kinematic Model 547
    14.4.1.3 Camera Projection Model 548
    14.4.1.4 Control Law 549
    14.4.1.5 Experimental Results 550
  14.4.2 Hybrid Controller for Robust Visual Servoing 554
    14.4.2.1 Control Scheme 554
    14.4.2.2 *Q*-Learning Controller 555
    14.4.2.3 Arbitrator of the Hybrid Control System 558
  14.4.3 Experimental Results 558
    14.4.3.1 Hybrid Visual Servoing with a Small Dangerous Area 558
    14.4.3.2 Hybrid Visual Servoing with a Large Dangerous Area 560
14.5 Conclusions 564
References 565

SUMMARY This chapter presents how to implement machine learning in robotic applications. In particular, reinforcement learning or *Q* learning is introduced and demonstrated in physical robotic tasks. The first case study shows how to integrate reinforcement learning with genetic algorithms to provide decision making for a multi-robot transportation system. The second case study demonstrates how to employ reinforcement learning to improve the reliability of visual servoing controllers. The corresponding simulation and experimental results are presented.

14.1 Introduction

Robotic systems usually work in a dynamic and unknown environment in which the traditional behavior-based or planning-based approaches can easily fail. In particular, for behavior-based approaches, it is very difficult to design a sufficiently extensive behavioral rule base to cope with all possible world states encountered by robots in a dynamic and unknown environment. Meanwhile, it is also challenging for a robot to plan its actions in advance when the robot environment is dynamic and changing fast.

By observing human capabilities in dealing with a dynamic environment, it is not difficult to draw a conclusion that a human employs not only planning or reactive (behavior-based) techniques but also learning techniques to successfully complete tasks. Through learning, the human brain learns new world states, identifies corresponding optimal actions from the past experience, and improves his or her planning and reactive techniques continuously. In other words, learning makes a human deal with unexpected world states, decreases uncertainties in the environment, and makes decisions that are more robust and effective in a dynamic environment.

Accordingly, machine learning has become an important topic in the robotics community. Especially in the multi-robot field, machine learning is regarded as a new and important trend due to the highly dynamic character of typical multi-robot environments.

Among the existing machine learning approaches, reinforcement learning (RL), especially Q learning, is used commonly in robotic systems due to its simplicity and good real-time performance. In this chapter, the Q learning algorithm will be introduced. Moreover, two case studies will be provided to show how to implement machine-learning algorithms in robotic systems.

14.2 Markov Decision Process (MDP) and Q Learning

The Markov decision process (MDP) is favored in the artificial intelligence (AI) community. Not only is it an important topic in the theoretical research of AI, it also has some practical applications, such as facilitating an intelligent agent to make decisions in a real-world situation. The basic idea of MDP is presented below.

MDP can be defined by the four-tuple $<S, A, T, R, \beta>$, where

- $S = \{s_1, s_2, \ldots, s_n\}$ is a set of states of the environment
- $A = \{a_1, a_2, \ldots, a_m\}$ is a set of actions available to the robot
- $T : S \times A \rightarrow \Pi(S)$ is a transition function, which decides the next environmental state s' when the robot selects an action a_i under the current state s. $\Pi(s)$ is the probability distribution over the states
- $R : S \times A \rightarrow \mathfrak{R}$ is a reward function, which determines the immediate reward when the robot takes an action a_i under the current state s

MDP describes a sequence of decisions made by an agent or a robot in an environment with many states. At time t, the agent needs to observe the environment and determine its current state $s \in S$, and then select an action $a \in A$ based on its policy π, which specifies what the agent should do for any state that it might reach. In addition, there is a reward function, which determines the immediate reward when the agent takes an action a under the current state s.

MDP is interesting and challenging for two reasons. The first one is that the subsequent state s' is not deterministic when the agent takes an action a under the current environmental state s. Instead, it has a probability distribution $\Pi(S)$ over all the states. This probability distribution is defined by a transition function or transition model $T(s, a, s')$. The second attribute of MDP is that the transitions among states are Markovian; that is, the probability of reaching s' from s depends only on s and not on the history of earlier states.

The performance of an agent policy π is measured by the rewards the agent received when it made a sequence of decisions according to this policy and visited a sequence of states. This measurement is usually represented by a sum of discounted rewards, as given by

$$\left[\sum_{t=0}^{\infty} \beta^t R(s_t) \mid \pi\right] \tag{14.1}$$

where, $0 < \beta \leq 1$ is the discount factor, and $R(s_t)$ is the reward received when the agent visits the state s_t at time t. Because the transitions among states are not deterministic in view of the probablistic nature of the transition function $T(s, a, s')$, given a policy, the sequence of states visited by the agent each time is not fixed and has a probability distribution. Therefore, an optimal policy π^* is a policy that yields the highest expected sum of the discounted rewards, which is given by

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \beta^t R(s_t) \mid \pi\right] \tag{14.2}$$

Given a MDP problem, one crucial consideration is how to find all optimal policies (if there exist more than one optimal policy). The value iteration algorithm has been developed to find the solution to this problem [1].

In the value iteration algorithm, first the concept of "utility of a state" is defined as

$$U(s) = E\left[\sum_{t=0}^{\infty} \beta^t R(s_t) \mid \pi^*, s_0 = s\right] \tag{14.3}$$

From Equation 14.3, the utility of a state s is given by the expected sum of discounted rewards when the agent executes the optimal policy.

If we have the utilities of all states, an agent's decision making (or action selection) will become easy, specifically, to choose the action that maximizes the expected utility of the subsequent state:

$$\pi^*(s) = \arg\max_{a} \sum_{s'} T(s, a, s') U(s') \tag{14.4}$$

However, the problem is how to find the utilities of all states. This is not an easy task. Bellman found that the utility of a state could be divided into two parts: the immediate reward for that state and the expected discounted utility of the next state, assuming that the agent chooses the optimal action.

$$U(s) = R(s) + \beta \max_{a} \sum_{s'} T(s, a, s')U(s') \tag{14.5}$$

Equation 14.5 is the famous Bellman equation. For each state, there is a corresponding Bellman equation. If there are n states in total, then we can have n Bellman equations. The utilities of the n states are found by solving the n Bellman equations. However, the Bellman equation is a nonlinear equation because it includes a "max" operator. The analytical solutions cannot be found using techniques of linear algebra. The only way to find the solutions of n Bellman equations is to employ some iterative techniques. The value iteration algorithm is such an approach to find the utilities of all states. This algorithm is presented below [1].

In Figure 14.1, the value iteration algorithm is introduced to solve MDP problems. It appears that MDP is no longer a difficult problem. However, in a real robot decision-making situation, the value iteration algorithm is not practical because the environment model $T(s, a, s')$ and the reward function $R(s)$ are usually unknown. In other words, it is usually impossible for one to obtain perfect information of the environmental model and employ the value iteration algorithm to solve the MDP problem. Reinforcement learning is developed to meet this challenge.

In reinforcement learning, through trials of taking actions under different states in the environment, the agent observes a sequence of state transitions and rewards received, which can be used to estimate the environmental model and approximate the utilities of the states. Therefore, reinforcement learning is a type of model-free learning, which is its most important advantage.

There are several variants of reinforcement learning among which Q learning is the most popular one. A typical Q-learning algorithm is presented below.

```
Function value-iteration (mdp, $\varepsilon$) returns a utility function
    Inputs: mdp, an MDP with state $S$, transition model $T$, reward function $R$, discount $\beta$, and the maximum
    error $\varepsilon$ allowed in the utility of any state.
    Local variables: $U$, $U'$ are the vectors of the utilities for the states in $S$, initially zero. $\delta$ is the maximum
    change in the utility of any state in an iteration.
    Repeat
        $U \leftarrow U'; \delta \leftarrow 0$
        For each state $s$ in $S$ do
            $U'(s) \leftarrow R(s) + \beta \max_{a} \sum_{s'} T(s,a,s')U(s')$
If $|U'(s) - U(s)| > \delta$    then    $\delta \leftarrow |U'(s) - U(s)|$
Until $\delta < \varepsilon(1 - \beta)/\beta$
Return $U$
```

FIGURE 14.1
Value iteration algorithm to calculate the utilities.

At the initiation of the Q-learning algorithm, an empty Q table is set up, and all its entries are initialized to zero. The Q table is a 2-D table in which the rows represent the environmental states and the columns represent the actions available to the agent (the robot). Here, the value of a Q table entry, $Q(s_i, a_j)$, represents how desirable it is to take the action a_j under the state s_i, and the utility $U(s_i)$ in Figure 14.1 represents how appropriate it is for the agent to be in the state s_i. Parameters such as the learning rate η, the discount factor β, and the "temperature" parameter τ, have to be initialized as well.

During operation, the agent observes the environment, determines the current state s, probabilistically selects an action a_k with probability given by Equation 14.6 and executes it.

$$P(a_k) = \frac{e^{Q(s,a_k)/\tau}}{\sum_{l=1}^{m} e^{Q(s,a_l)/\tau}} \tag{14.6}$$

After the agent takes the action a_k, it will receive a reward r from the environment and observe the new environment s'. Based on the information of r and s', the agent will update its Q table according to

$$Q(s,a_k) = (1-\eta)Q(s,a_k) + \eta(r + \beta \max_{a'} Q[s',a']) \tag{14.7}$$

In this manner, the current environmental state is transitioned from s to s'. Based on the new state, the above operations are repeated until the values of the Q table entries converge.

In the Q-learning algorithm described in Figure 14.2, an ε – *greedy* search policy presented in Equation 14.6 is used instead of a greedy policy, which always selects the action with the maximum Q value. In an ε – *greedy* search policy, with probability ε, the agent chooses one action uniformly randomly among all possible actions, and with probability $1 - \varepsilon$, the agent chooses the action with a high Q value under the current state. In addition, the probability ε is decreased gradually. The advantage of the ε – *greedy* search policy is its balance in exploring unknown states against exploiting known states when the agent

```
• For each state $s_i \in (s_1,s_2,\ldots,s_n)$ and action $a_j \in (a_1,a_2,\ldots,a_m)$, initialize the table entry $Q(s_i, a_j)$ to zero.
  Initialize $\tau$ to 0.9. Initialize the discount factor $0 < \beta \le 1$ and the learning rate $0 < \eta \le 1$.
• Observe the current state $s$
• Do repeatedly the following:
    • Probabilistically select an action $a_k$ with probability
      $P(a_k) = \frac{e^{Q(s,a_k)/\tau}}{\sum_{l=1}^{m} e^{Q(s,a_l)/\tau}}$, and execute it
    • Receive the immediate reward $r$
    • Observe the new state $s'$
    • Update the table entry for $Q(s,a_k)$ as follows:
    • $Q(s, a_k) = (1 - \eta)Q(s, a_k) + \eta(r + \beta\max Q[s',a'])$
    • $s \leftarrow s'$, $\tau \leftarrow \tau * 0.999$
```

FIGURE 14.2
Single-agent Q learning algorithm.

attempts to learn its decision-making skills and improve its Q table. In Equation 14.6, the probability ε is controlled through decreasing the "temperature" parameter τ.

14.3 Case Study: Multi-Robot Transportation Using Machine Learning

An important research topic in multi-agent robotics is multi-robot object transportation. There, several autonomous robots move cooperatively to transport an object to a goal location and orientation in a static or dynamic environment, possibly avoiding fixed or removable obstacles. It is a rather challenging task. For example, in the transportation process, each robot needs to sense any change in the environment, the positions of the obstacles, and the other robots in the neighborhood. Then it needs to communicate with its peers, exchange the sensing information, discuss the cooperation strategy, suggest the obstacle avoidance strategy, plan the moving path, assign or receive the subtasks and roles, and coordinate the actions so as to move the object quickly and successfully to the goal location. Arguably, the success of this task will require the use of a variety of technologies from different fields. As a result, the task of multi-robot transportation is a good benchmark to assess the effectiveness of a multi-agent architecture, cooperation strategy, sensory fusion, path planning, robot modeling, and force control. Furthermore, the task itself has many practical applications in fields such as space exploration, intelligent manufacturing, deep sea salvage, dealing with accidents in nuclear power plants, and robotic warfare.

In this section, a physical multi-robot system is developed and integrated with machine learning and evolutionary computing for carrying out object transportation in an unknown environment with simple obstacle distribution. In the multi-agent architecture, evolutionary machine learning is incorporated, enabling the system to operate in a robust, flexible, and autonomous manner. The performance of the developed system is evaluated through computer simulation and laboratory experimentation.

As explained in Section 14.1, a learning capability is desirable for a cooperative multi-robot system. It will help the robots to cope with a dynamic or unknown environment, find the optimal cooperation strategy, and make the entire system increasingly flexible and autonomous. Although most of the existing commercial multi-robot systems are controlled remotely by a human, autonomous performance will be desirable for the next generation of robotic systems. Without a learning capability, it will be quite difficult for a robotic system to become fully autonomous. This provides the motivation for the introduction of machine-learning technologies into a multi-robot system.

The primary objective of the work presented here is to develop a physical multi-robot system, where a group of intelligent robots work cooperatively to transport an object to a goal location and orientation in an unknown and dynamic environment. A schematic representation of a preliminary version of the developed system is shown Figure 14.3.

14.3.1 Multi-Agent Infrastructure

A multi-agent architecture is proposed in Figure 14.4 as the infrastructure to implement cooperative activities between robots.

In Figure 14.4, four software agents and two physical agents are shown in the developed architecture, forming the overall multi-agent system. Each agent possesses its own internal state (intention and belief) and is equipped with independent sensing and decision-making

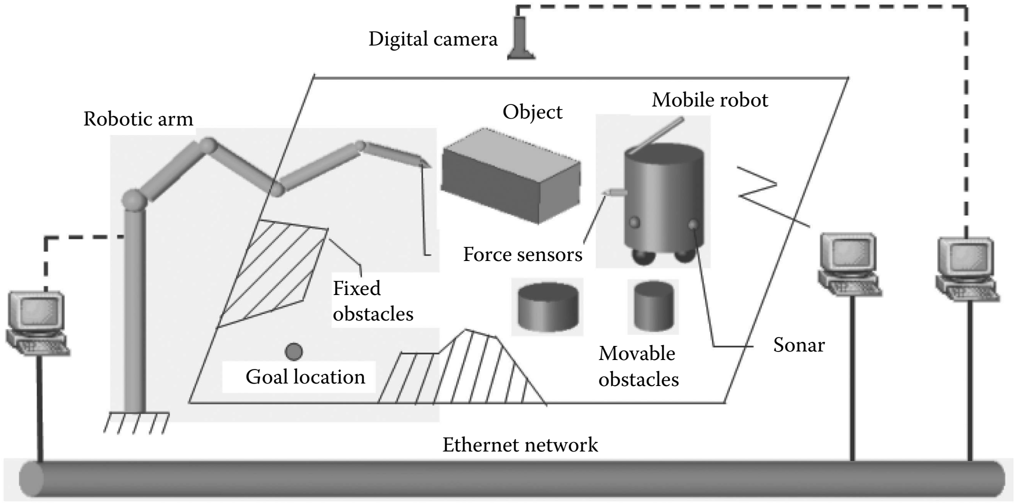

Digital camera
Robotic arm
Object
Mobile robot
Force sensors
Fixed obstacles
Goal location
Movable obstacles
Sonar
Ethernet network

FIGURE 14.3
The developed multi-robot system.

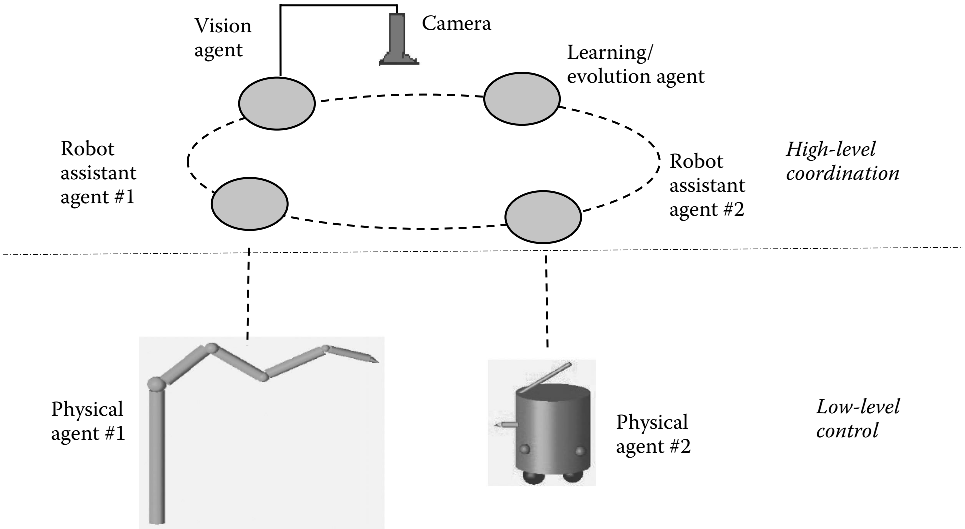

Vision agent
Camera
Learning/ evolution agent
Robot assistant agent #1
Robot assistant agent #2
High-level coordination
Physical agent #1
Physical agent #2
Low-level control

FIGURE 14.4
Multi-agent architecture used in the developed system.

capabilities. They also are able to communicate with each other and exchange information on their environment as acquired through sensors and the intentions and actions of its peers. Based on the information from their own sensors and their internal states, the agents cooperatively determine a cooperation strategy to transport the object.

In Figure 14.4, the four software agents constitute a high-level coordination subsystem. They will cooperate and coordinate with each other to generate cooperation strategies, and assign subtasks to the two robot assistant agents. In the meantime, the two physical agents (robots) form a low-level subsystem of control and execution, which receives commands from the upper-level subsystem.

There is a vision agent, which is in charge of acquiring and processing the images using a CCD camera. Because the present section focuses on developing a learning-based multi-robot architecture, a simplified computer vision scheme is employed, which uses a global camera and a global localization technology. The vision agent will analyze the acquired images and compute the exact locations and orientations of the robots, the object, and the obstacles. All the information is then broadcast to the other three software agents so that they will immediately know the current world state (the position and orientation messages of the robots, the object, and the obstacles and the environmental map information).

A special learning and evolution agent is used in the architecture to play the combined role of a learner, a monitor, and an advisor. The machine-learning algorithm and the decision-making capabilities are embedded in this agent. First, it will collect from other agents the information on the environment, the robots, the object, and the obstacles to form the current world state, which can be shared by its peers. Then it will make decisions on the optimal cooperation strategy based on its own knowledge base so that the common task can be completed effectively and quickly. Finally, it will assign the roles of other agents, and monitor the task schedule.

Each physical agent (robot) is linked to a corresponding assistant agent, which needs to forward the robot position/force information to the other agents in the upper level, to receive the desired force/trajectory of the robot from the learning/evolution agent, and to send them to the corresponding robot in the low-level subsystem. All six agents will form a tightly coupled multi-agent system to transport the object cooperatively to the goal location while avoiding obstacles.

14.3.2 Cooperation Based on Machine Learning

Learning ability is a desirable feature for a multi-robot transportation system. It will help the robots to cope with dynamic changes in an unknown environment. Although conventional planning algorithms can make the robots move effectively in a static environment, they usually fail in an environment with dynamic obstacles. A multi-robot system with learning capability possesses characteristics of adaptation and self-organization, which will enable it to overcome the difficulties in an unknown environment. In this section, two well-known machine-learning approaches, RL and genetic algorithms (GAs), are integrated into the learning and evolution agent to achieve robust and effective cooperation between robots.

14.3.2.1 Reinforcement Learning

A key problem in a multi-robot transportation system is the determination of the cooperation strategy, that is, finding the optimal amplitudes and locations of the applied forces of the robots so that a net force with maximum amplitude will point toward the goal location while avoiding any obstacles. RL provides a way to determine an appropriate cooperation strategy through experimentation with the environment. A method of reinforcement learning known as the Q-learning algorithm, as given in Figure 14.2, is employed here to find the optimal action policy for each robot.

In the system developed in this section, the world state is defined as the relative position and orientation between the object and the goal location and the positions of the obstacles within a fixed detection radius around the object. This is shown in Figure 14.5.

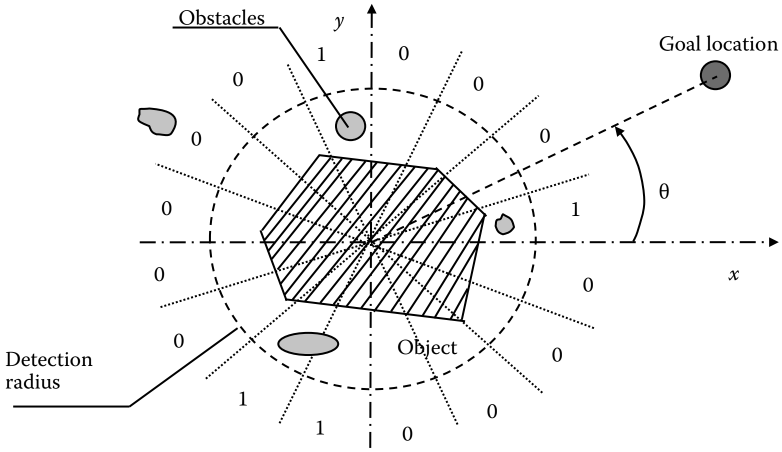

Obstacles
y
1
0
0
0
Goal location
0
0
0
1
θ
0
x
0
0
Object
0
Detection radius
1
0
1
0

FIGURE 14.5
The world state representation.

In Figure 14.5, only the obstacles located within the detection radius around the object are detected and considered. The detection circle is evenly divided into 16 sectors, and the sectors are represented by a 16-bit binary code. If one obstacle is located in one of the sectors, the corresponding bit of the binary code will become "1." Otherwise it will become "0." Accordingly, this binary code represents the obstacle distribution around the object. The world state s in the Q-learning algorithm is made up of this binary code and the angle θ in Figure 14.5, which describes the relative orientation between the goal location and the object. For example, the world state shown in Figure 14.5 can be represented by the vector $s = [\theta\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0]$.

The cooperation strategies, that is, the force locations and amplitudes for all robots, which are shown in Figure 14.6, form the set of actions in the Q-learning algorithm.

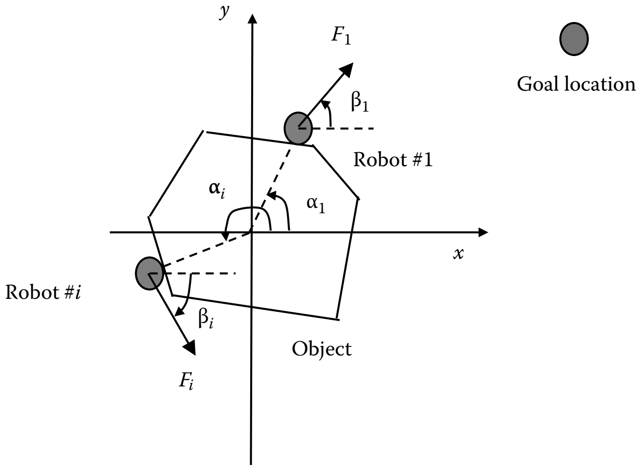

y
F_1
Goal location
β_1
Robot #1
α_i
α_1
x
Robot #i
β_i
Object
F_i

FIGURE 14.6
Representation of the cooperation strategy (action).

In Figure 14.6, the application point of the force from each robot is represented by the pair of angles (α, β). Therefore, possible actions in the Q-learning algorithm for the robot R_i can be modeled as the set $\Lambda_i = \{(\alpha, \beta, F)_k\}$ $k = 1, 2, \ldots, m_i$.

There remains a special action called *sweeping*, which instructs the robots to remove all the obstacles in the path after establishing that they are capable of carrying out this sweeping task.

The reward r is calculated as

$$r = D_{t_i} - D_{t_{i-1}} \tag{14.8}$$

where, D_{t_i} and $D_{t_{i-1}}$ are the distance between the mass center of the object and the goal location, at times t_i and t_{i-1}, respectively.

14.3.2.2 Genetic Algorithms

Another approach that is used here to find the optimal cooperation strategy is GAs. By simulating the biological evolution process, GA provides a powerful way to quickly search and determine the optimal individual in a large candidate space.

A typical genetic algorithm as found in [2] is given below.

In this section, the entire candidate space is formed by the *sweeping* action and $\Phi = (\alpha_1, \beta_1, F_1, \cdots, \alpha_i, \beta_i, F_i, \cdots \alpha_n, \beta_n, F_n)$ described in Figure 14.6, which is the set of all possible cooperation strategies. This candidate space can be represented by a vector space $\mathfrak{R}^{3n+1}$, which is made up of all possible actions: (Φ, *sweeping*). The genetic algorithm given in Figure 14.7 is used to search for the optimal individual vector, that is, the optimal robot cooperation strategy, in this large vector space.

In the beginning, the *sweeping* action and nine individuals randomly selected from the candidate space are used to constitute the first generation of population. Then the genetic

```
• Initialize $p$ (the number of individuals in the population), $\eta$ (the fraction of the population to be replaced by
  crossover at each step), $m$ (the mutation rate), and fitness threshold.
• Generate $p$ individuals at random to produce the first generation of population $P$
• Calculate the fitness($i$) for each $i$ (individual) in $P$
• While $(\max_i \text{fitness}(i)) <$ fitness_threshold, then repeat:
  Produce a new generation $P_s$:
  1. Probabilistically select $(1 - \boldsymbol{\eta})\boldsymbol{p}$ members of $P$ to be included in $P_s$. The selection probability is given by:
     $\lambda(i) = \dfrac{\text{Fitness}(i)}{\sum_{j=1}^{p} \text{Fitness}(j)}$
  2. Execute the crossover operation to generate $(\boldsymbol{\eta p})/2$ pairs of offspring and add them to $P_s$
  3. Mutate $m$ percent of the numbers of $P_s$
  4. Update $P \leftarrow P_s$
  5. Calculate fitness($i$) for each $i$ in $P$
• Return the individual with the highest fitness in $P$
```

FIGURE 14.7
The genetic algorithm.

algorithm outlined previously is used to search for the optimal individual with the maximum fitness value. The fitness is calculated as follows:

$$\text{Fitness} = \begin{cases} C & \text{if it is a ``}\textit{sweeping}\text{'' action and some} \\ & \text{removable obstacles are in the path.} \\ O & \text{if a ``}\textit{sweeping}\text{'' action without} \\ & \text{removable obstacles in the path.} \\ M & \text{otherwise} \end{cases} \tag{14.9}$$

$$M = \frac{k_1 * |F| + k_2 * (1+\cos\theta)^2 + k_3 / |\Gamma| + k_4 S + k_5 / (1+\cos\tau)^2}{k_1 + k_2 + k_3 + k_4 + k_5} \tag{14.10}$$

where C is a threshold value, which is determined through trials, F is the net force applied by the robots, θ is the angle between the net force vector and the goal location vector, which is described in Figure 14.5, Γ is the net torque applied by the robots, S is the area of the polygon formed by the robot locations, τ is the angle between the net force vector and the position vector of its closest obstacle, and k_1 to k_5 are the weights ($k_1 + k_2 + k_3 + k_4 + k_5 = 1$).

The fitness function includes several practical considerations. First, the *sweeping* action is always intended for use in the current world state. If a removable obstacle is present in the forward path of the object (it is assumed that, based on its size, color, shape, and so on, the agents are able to judge whether the obstacle can be moved away by the robots), the *sweeping* action will assume a high fitness value so that the obstacle can be removed immediately. In order for it to always have the opportunity to be selected, the *sweeping* action is included in the population of the first generation, and it is always copied to the next generation in the evolution process.

Second, a large net force that points to the goal location will earn a high fitness value so that the internal force becomes a minimum and the object is transported to the goal location quickly and effectively. Third, a high net torque Γ is not encouraged because the resulting unnecessary rotation of the object will make it difficult for the robots to handle it, and a small Γ may be encouraged for avoiding the obstacles. Fourth, a large S implies that the spacing of the robots is quite adequate, which will receive a high fitness value, because crowding of the robots will make their operations more difficult and less effective. Finally, a small τ means that the probability of the presence of an obstacle in the path is high, and this condition should be punished. The weights $k_1 \sim k_5$ are calibrated through experimentation.

In each step of object transportation, the GA is used to search for the optimal cooperation strategy. Then this strategy is broadcast to the two robot assistant agents by the learning and evolution agent. Next, the robot assistant agents forward the commands to the physical robots. Based on their internal hybrid force/position controllers, the robots then move themselves to the desired positions and transport the object for a specified time period. In the meantime, the current force and position information is sent back to the high-level coordination subsystem via the robot assistant agents for decision making in the next step. After the object is moved through some distance, the world state would be changed (for example, new obstacles are observed and so on). In this manner, the GA is used again to search for the new optimal cooperation strategy under the new world state. This process will be repeated until the object is successfully moved to the goal location.

The main problem of GA in a physical robotic system is that its real-time performance cannot be guaranteed because of its low convergence speed. However, in the present project, this problem is solved partly by using the layered architecture shown in Figure 14.4. In this two-layer multi-agent architecture, the low-level physical agents can continue to execute the old commands before the new commands are given by the agents in the upper level. In other words, the agents in the upper level are not necessary to work at the speed of the physical agents in the lower level. This enables a "low-speed" GA agent to make decisions for a real-time task.

14.3.2.3 Integrating Reinforcement Learning with Genetic Algorithms [3]

RL may face difficulties and generate nonoptimal outputs. A typical problem is that "the agent runs the risk of over-committing to actions that are found to have high Q values during the early stages of training, while failing to explore other actions that also have high values" [4]. Another problem is that RL may not be able to capture the slow changes in the environment. In other words, when the environment changes very slowly, RL will have less opportunity to try the actions under different world states. Consequently, the knowledge base of RL will be updated very slowly. This means that the learning process will become very slow and the output of RL will not be optimal.

GAs can partly overcome these problems. Because of the special evolution operations, such as crossover and mutation in GAs [2], it is possible for GAs to select a new search point in the candidate space that had not been tried in the past steps. This will provide an opportunity for the actions with low Q values in reinforcement learning to demonstrate their capabilities. Moreover, because a GA simulates the biological evolution process, it provides better adaptability than RL and is able to capture slow changes in the environment.

On the other hand, GAs also benefit from RL because GAs cannot guarantee a precisely optimal output. In particular, in the system developed in the present work, the crowding phenomenon [4] was observed, where "very similar individuals take over a large fraction of the population." In addition, GAs are unable to deal with quick changes in the environment because the evolution process usually requires a considerable duration of time. All these problems can greatly slow down the evolution process and make the output of the GAs unacceptable.

However, by integrating GAs with RL, different individuals can be inserted into the population of GAs through RL because RL uses an entirely different mechanism to select the new cooperation strategy. Consequently, it would be possible to disturb the population purity in a GA and resolve the "crowding" problem in part. In addition, RL is able to deal with quick changes in the environment, and GA usually fails.

The combination of learning with evolution, two important adaptive mechanisms that exist in nature, gives rise to distinct advantages as discussed previously. In this section, a specific integration scheme of these two capabilities is implemented in the multi-agent object transportation system, which is indicated in Figure 14.8.

In Figure 14.8, the sensor data is first sent to a sensory fusion module. This information will form the current positions and orientations of the object, the robots, and the obstacles and the state of the environment. Moreover, there is a model knowledge base in which the geometry and dynamic parameters of the object and the robots are stored. These model

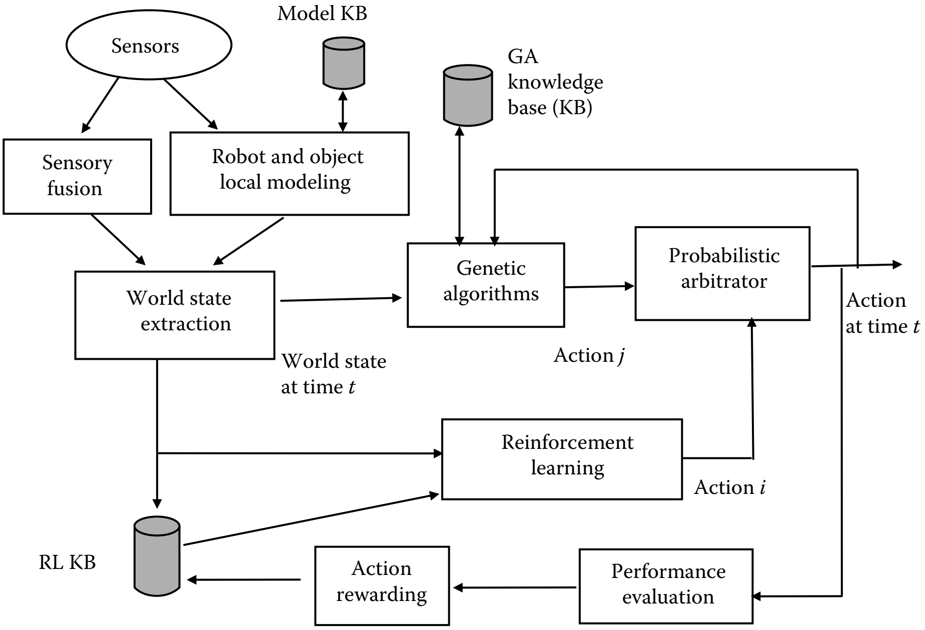

Sensors
Model KB
GA knowledge base (KB)
Sensory fusion
Robot and object local modeling
World state extraction
Genetic algorithms
Probabilistic arbitrator
World state at time *t*
Action *j*
Action at time *t*
Reinforcement learning
Action *i*
RL KB
Action rewarding
Performance evaluation

FIGURE 14.8
Integration scheme of RL and GA.

parameters are determined off-line and stored in the model knowledge base prior to operating the system. When the sensory data is sent to a local modeling module, the module will identify and update the current model parameters by combining the off-line information in the knowledge base and the new sensory data. The information from the sensory fusion module and the model knowledge base is then integrated to form the new world state as described in Figure 14.5.

The two decision-making mechanisms, reinforcement learning and genetic algorithm, are implemented based on the algorithms presented in the previous sections. They both are able to determine a robot cooperation strategy according to the current world state. Their outputs are probabilistically selected by an arbitrator, which determines the winning output. In this chapter, it is assumed that the output of the reinforcement learning block is selected by the arbitrator with a probability of 0.7, and the output of the genetic algorithm block is selected with a probability of 0.3. The output of the arbitrator, that is, the cooperation strategy, is then broadcast to the two robot assistant agents by the learning and evolution agent to implement the object transportation task.

14.3.3 Simulation Results

Some simulation work has been carried out to demonstrate the performance of the evolutionary learning mechanism described previously. Because the simulation is used to validate the evolutionary learning approach in Section 14.3.2, a special case is assumed, which is somewhat different from Figure 14.4. Specifically, it is assumed that there are three autonomous robots, which cooperate to transport a rectangular box (object) to a goal location in a 400 × 400-unit environment without any obstacles. The box has a width of 60 units and a height of 40 units. Each robot is simplified as a circular point in the environment shown in the subsequent figures. The multi-thread programming of Java language is used to simulate the parallel operations of the three robots.

The present work is applicable not only in object transportation tasks with small displacements described in Figure 14.4, but also in more challenging cases, such as space exploration. For example, consider the case of several robots sent to a planet for exploration of a special ore, and they transport the ore cooperatively into a spacecraft. In this application, it is impossible for a robot or an agent to possess global information. In other words, each robot only possesses the local sensing ability, and it will need to communicate with the other robots to establish the global information. In the present simulation, each robot is assumed to be only equipped with local sensing capability. Moreover, the evolutionary learning mechanism is also embedded in each robot so that they can cope with the possible existence of unreliable communication in planetary exploration.

Because a robot only possesses local sensing capability and the environment dimension is somewhat large, it will not know where the box is in the beginning. Therefore, a random search strategy is used to find the object in the beginning of the simulation. Specifically, the robots walk randomly to search the object. When one robot reaches a wall or another robot, it simply changes its direction and continues the "wandering state." However, if a robot reaches the box, the "wandering" behavior ceases and the position of the box is immediately broadcasted to the peers. Then this robot will temporarily take over the "leader" role in the subsequent transportation task. Next, it will use its embedded evolutionary learning mechanism to compute the optimal cooperation strategy and send it out to the other robots. Consequently, the other robots will move themselves to the desired locations and orientations, which are described in the new cooperation strategy sent by the leader robot. This process is shown in Figure 14.9.

The leader robot is responsible for synchronizing the forces and motions among the teammates. It will check whether all the robots have moved to the desired position and

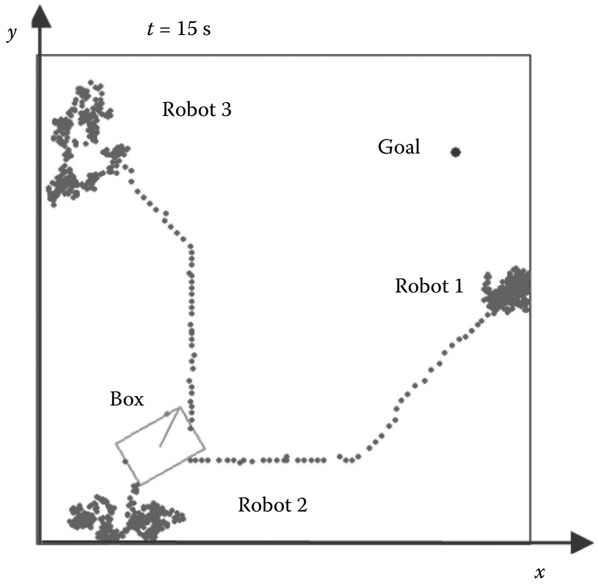


FIGURE 14.9
Robots move randomly and locate the box.

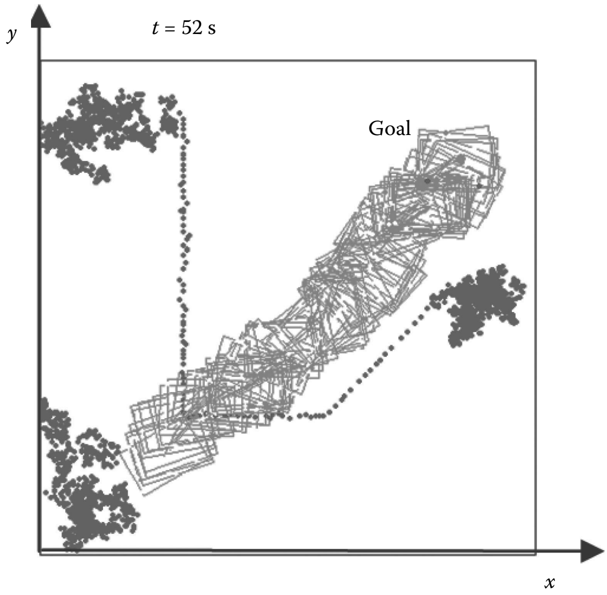


FIGURE 14.10
The whole transportation process.

orientation. If all the robots are ready, the leader robot will send out a "transport" command. Then the robots will exert the desired forces on the box, and the position and orientation of the box will be changed as a result. Next, the leader robot will compute the new world state and generate the new cooperation strategy. This process will be repeated until the box is transported to the goal location and orientation. The overall process, as simulated, is shown in Figure 14.10.

Figure 14.11 demonstrates the adaptability of the evolutionary learning mechanism employed by the robots. In this example, the goal location was suddenly changed from (380,370) to (385,60) at $t = 34$ sec while the robots were transporting the box. From Figure 14.11, it is observed that the robots quickly adjusted the cooperation strategy and successfully transported the box to the new goal location.

From the simulation results presented here, it is seen that the evolutionary learning mechanism described before is quite effective in meeting its design objective.

14.3.4 Experimentation

The experiment presented here is designed to test the multi-agent learning strategy described before. Here, a movable obstacle is placed in the path while the robots are cooperatively transporting the box to the goal location. Figure 14.12 shows how the robots handle this problem.

From Figure 14.12, it is seen that when the robots detect the existence of a movable obstacle in the forward path with the help of the camera agent, they select the "sweeping" action. Accordingly, the mobile robot leaves the box, moves toward the obstacle, and pushes it away from the path. While the mobile robot is sweeping the obstacle, the robotic arm just stays in the same place, waiting for its peer to return. Once the obstacle is swept

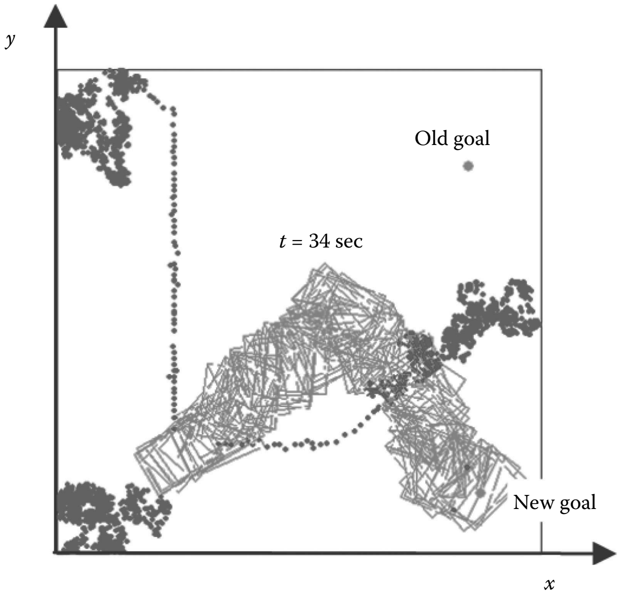
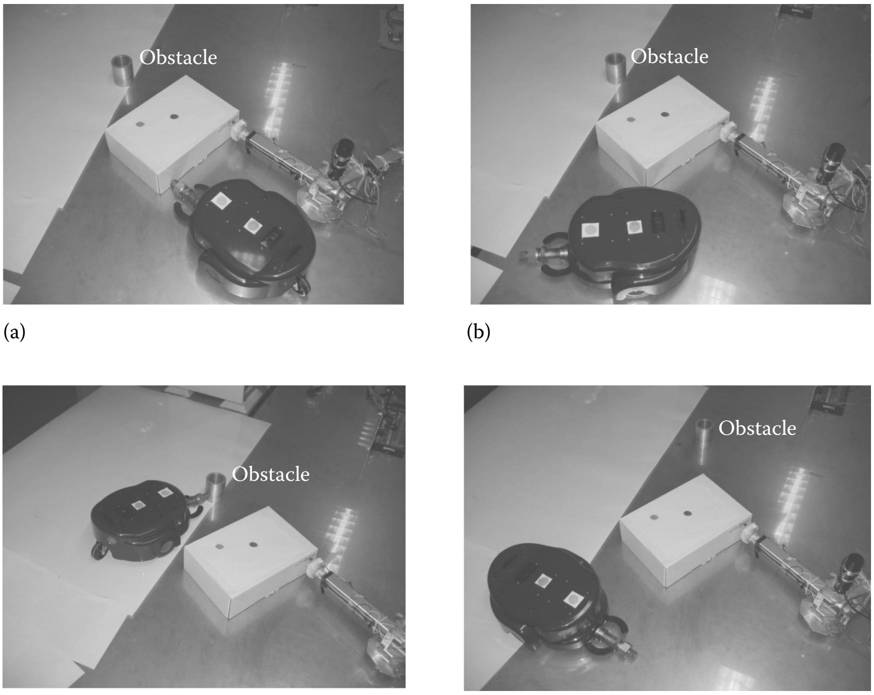


FIGURE 14.11
The goal location was suddenly changed from (380,370) to (385,60) at t = 34 sec while the robots were transporting the box.


Obstacle
Obstacle
Obstacle
Obstacle

(a) (b) (c) (d)

FIGURE 14.12
The robots have selected the "sweeping" action to remove a movable obstacle in the path. (a) Initial state (pushing of the box, (b) leaving the box, (c) removing the obstacle, and (d) returning.

away, the mobile robot returns to continue the task of box transportation in cooperation with the robotic arm.

This experiment demonstrates that the multi-agent robotic system developed in the present work, equipped with machine-learning capability, exhibits good flexibility and adaptability in a complex environment while effectively carrying out the intended task.

14.4 Case Study: A Hybrid Visual Servo Controller Using *Q* Learning

A multi-robot cooperative assembly project is being developed. In this project, multiple mobile robots autonomously search for the parts of a target vehicle, which are scattered in an unstructured environment with complex terrain and obstacle distribution. Once an object of interest is encountered and identified, a robot will attempt to autonomously grasp it and pick it up with its onboard arm using visual servoing and transport it to a designated site for further manipulation and assembly. When the robots determine that they have collected all the necessary parts, they proceed to assemble them to construct the target vehicle, by cooperatively manipulating the parts.

In the present section, a core requirement of the project is accomplished. Specifically, a robust scheme of visual servo control is developed for driving a robot toward a target object and for autonomously grasping and manipulating the object. The visual controller developed here will integrate the capabilities of the mobile robot base, the onboard arm, and the CCD camera. A Pioneer™ DX3 mobile robot with a 5 degree of freedom (5DOF) onboard arm is used in the present work as shown in Figure 14.13.

The mobile robot in Figure 14.13 is a non-holonomic vehicle with two drive wheels whose speeds can be controlled independently and can be measured using the 500-pulse encoders mounted on them. A color CCD camera and a 5DOF Pioneer Arm™ are mounted on the mobile robot to constitute a vision-based mobile manipulation system. The onboard Pioneer Arm™ is a 5DOF robot with a gripper consisting of foam-lined fingers for a firm

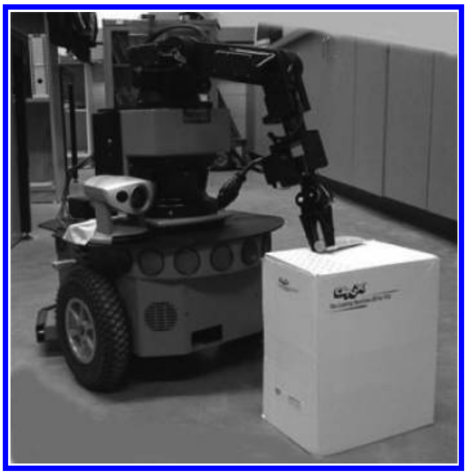


FIGURE 14.13
The experimental setup of visual servo control.

grasp. Driven by six reversible open-loop servo motors, the tip of the closed fingers of the arm has a reach of 50 cm from the center of the rotating base. The color CCD camera is a pan-tilt-zoom (PTZ) vision system with 26× optical and 12× digital zoom for a wide range of applications.

In addition, the robot comes with ACTS™ color blob–tracking software, which is employed to track the image coordinates of the target object, which in Figure 14.13 is a toothpaste tube with a red cap.

In addition to the Pioneer Arm and the CCD camera, the robot is equipped with an onboard computer and numerous sensors, such as a laser distance finder, sonar ring, compass, and a gyro. The onboard computational capability can support real-time performance of the visual servo control system.

Unlike the classical visual servo control approach [5], the control scheme developed in this section decouples the control objective between the degrees of freedom of the mobile robot base and the onboard 5DOF arm. Specifically, the control process is decoupled into two steps when the robot attempts to pick up an object of interest in its field of view. In the first step, by observing the visual error between the current image and the prerecorded image at the desired location and orientation, the mobile robot base attempts to change its position and heading by adjusting the angular velocities of its wheels so as to make the robot base move closer to the target object and align itself with the object. In the second step, when the mobile robot base moves sufficiently close to the target object, the onboard arm reaches and grasps the object using the image-based eye-to-hand visual servo control scheme as described in [5]. Because the approach used in the second step is well known, the present section focuses primarily on the development of the visual servo control law in the first step.

14.4.1 Vision-Based Mobile Robot Motion Control

14.4.1.1 Control Scheme

In this section, the classical image-based visual servoing (IBVS) approach for fixed-base manipulators is extended for the motion control of wheeled mobile robots (WMR) as schematically shown in Figure 14.14.

In Figure 14.14, a CCD camera is mounted on the mobile base, which is able to continuously capture live images of the target object. The captured image is compared with the prerecorded desired image to determine the visual errors. Using the visual errors, the IBVS controller computes the desired speeds ω_1, ω_2 of the wheels of the mobile robot, and sends them to the low-level proportional–integral–derivative (PID) controller of the robot. The robot then rotates and translates continuously until it reaches the desired position and orientation at which the camera will observe the desired image.

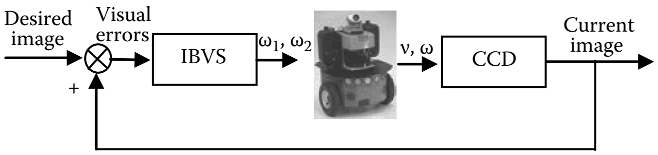


FIGURE 14.14
Visual servoing scheme for motion control of wheeled mobile robots.

14.4.1.2 Kinematic Model

Four sets of coordinate frames are defined now, namely the robot frame, the camera frame, the image plane frame, and the pixel coordinate frame, in order to derive the IBVS control law for the mobile robot. The relationship between the first two coordinate frames is shown in Figure 14.15.

In Figure 14.15, the camera frame is rigidly attached to the camera while the robot frame is fixed to the mobile robot. The coordinate transformation between the two frames is given by

$$\mathbf{H}_c^r = \begin{bmatrix} \mathbf{R}_c^r & \mathbf{d}_c^r \\ 0 & 1 \end{bmatrix} \tag{14.11}$$

Here, $\mathbf{H}_c^r$ represents the homogeneous transformation from the camera frame to the robot frame, $\mathbf{R}_c^r$ is the rotational matrix, and $\mathbf{d}_c^r = [d_x d_y d_z]^T$ is the position vector of the origin of the camera frame with respect to the robot frame. Because the camera is rigidly attached to the mobile robot base, the relationship between the camera velocity and the robot velocity may be obtained from Equation 14.12 as [5]

$$\xi_r^r = \begin{bmatrix} 0 \\ 0 \\ v \\ 0 \\ \omega \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^r & \mathbf{s}\left(\mathbf{d}_c^r\right) \mathbf{R}_c^r \\ \mathbf{0}_{3\times 3} & \mathbf{R}_c^r \end{bmatrix} \xi_c^c = \mathbf{G}\xi_c^c \tag{14.12}$$

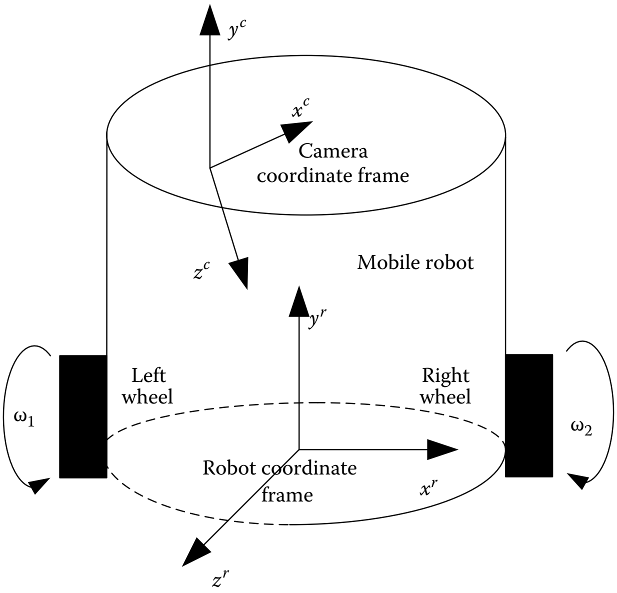


FIGURE 14.15
Robot frame and camera frame.

Here $\boldsymbol{\xi}_r^r$ is the robot velocity vector with respect to the robot frame. Also, v and ω are the translational and rotational velocities of the robot on the ground. In addition, $\boldsymbol{\xi}_c^c = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ w_z]^T$ is the camera velocity vector with respect to the camera frame, and $\mathbf{s}\left(\mathbf{d}_c^r\right)$ is the skew symmetric matrix of the vector $\mathbf{d}_c^r$.

The translational and rotational velocities of the robot can be expressed in terms of the wheel speeds as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} D/2 & D/2 \\ D/l & -D/l \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{14.13}$$

Here, D is the wheel diameter, l is the distance between the two wheels, and ω_1 and ω_2 are the speeds of the wheels.

14.4.1.3 Camera Projection Model

The relationship between the camera frame, the image plane frame, and the pixel coordinate frame is represented as in Figure 14.16.

In Figure 14.16, P is a point in the work environment with coordinates $(x, y, z)^c$ relative to the camera frame, and p is the projection of P on the image plane with coordinates (u, v) relative to the image plane frame and with coordinates (r, c) relative to the pixel coordinate frame. The distance between the origin of the camera frame and the image plane is denoted by λ, and the coordinates of the principal point is (o_r, o_c) with respect to the pixel coordinate frame. The coordinate transformation between the frames is given by [5]

$$\begin{cases} u = -s_x(r - o_r) \\ v = -s_y(c - o_c) \end{cases} \tag{14.14}$$

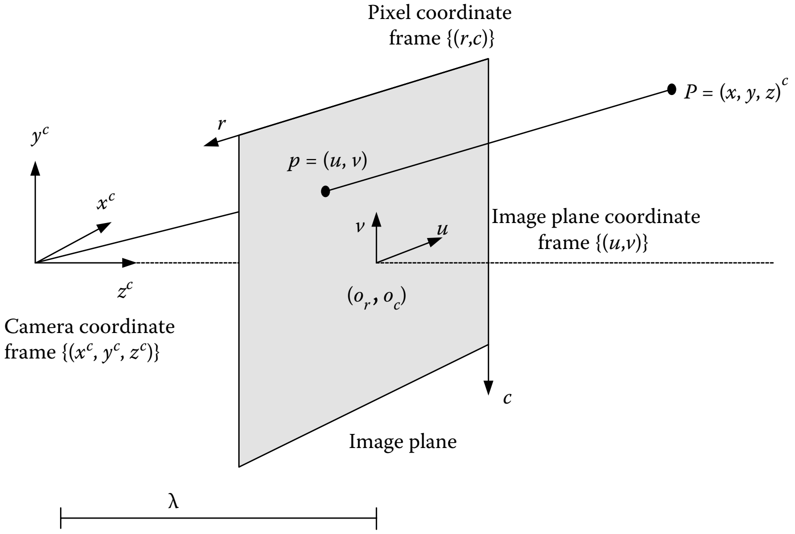

Pixel coordinate frame {(r,c)}
P = (x, y, z)^c
r
y^c
p = (u, v)
x^c
v
u
Image plane coordinate frame {(u,v)}
z^c
(o_r, o_c)
Camera coordinate frame {(x^c, y^c, z^c)}
c
Image plane
λ

FIGURE 14.16
Relationship between the camera frame, the image coordinate frame, and the pixel coordinate frame.

$$\begin{cases} r = -f_x \dfrac{x^c}{z^c} + o_r \\ c = -f_y \dfrac{y^c}{z^c} + o_c \end{cases} \tag{14.15}$$

Here, s_x and s_y are the horizontal and vertical dimensions, respectively, of a pixel as given by $f_x = \dfrac{\lambda}{s_x}$ and $f_y = \dfrac{\lambda}{s_y}$.

Let p be a feature point on the image plane with coordinates (u, v). The moving velocity of p can be expressed in terms of the camera velocity using the interaction matrix $\mathbf{L}$ as [5]:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \mathbf{L}\boldsymbol{\xi}_\mathrm{c}^\mathrm{c} = \begin{bmatrix} -\dfrac{\lambda}{z^c} & 0 & \dfrac{u}{z^c} & \dfrac{uv}{\lambda} & -\dfrac{\lambda^2 + u^2}{\lambda} & v \\ 0 & -\dfrac{\lambda}{z^c} & \dfrac{v}{z^c} & \dfrac{\lambda^2 + v^2}{\lambda} & -\dfrac{uv}{\lambda} & -u \end{bmatrix} \boldsymbol{\xi}_\mathrm{c}^\mathrm{c} \tag{14.16}$$

In the present work, the depth information z^c in the interaction matrix (Equation 14.16) can be estimated using the onboard laser/sonar distance finder of the mobile robot in real time.

14.4.1.4 Control Law

In this section, an image-based eye-in-hand visual servo control law is developed for motion control of a wheeled mobile robot. The underlying concept is that the controller will continuously adjust the wheel speeds of the mobile robot so that the coordinates (u, v) of the feature point are moved toward the desired position (u_d, v_d) on the image. In particular, the error vector of the image feature point is defined as

$$\mathbf{e} = \begin{bmatrix} u - u_d \\ v - v_d \end{bmatrix} = \begin{bmatrix} -s_x(r - r_d) \\ -s_y(c - c_d) \end{bmatrix} \tag{14.17}$$

$$\dot{\mathbf{e}} = \begin{bmatrix} \dfrac{\mathrm{d}(u - u_d)}{\mathrm{d}t} \\ \dfrac{\mathrm{d}(v - v_d)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \tag{14.18}$$

By substituting Equations 14.16 and 14.12 into Equation 14.18, one obtains

$$\dot{\mathbf{e}} = \mathbf{L}\boldsymbol{\xi}_\mathrm{c}^\mathrm{c} = \mathbf{L}\mathbf{G}^{-1}\boldsymbol{\xi}_\mathrm{r}^\mathrm{r} = \mathbf{M}\begin{bmatrix} v \\ \omega \end{bmatrix} \tag{14.19}$$

Here, **M** is a 2 × 2 matrix, which is constituted by the third and the fifth columns of $\mathbf{LG}^{-1}$. By substituting Equation 14.13 into Equation 14.19, we get

$$\dot{\mathbf{e}} = \mathbf{M}\begin{bmatrix} D/2 & D/2 \\ D/l & -D/l \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{14.20}$$

Equation 14.20 shows the relationship between the error rate defined on the image plane and the angular velocities of the wheels of the mobile robot. In view of Equation 14.20, assuming the error dynamics as $\dot{\mathbf{e}} = -k\mathbf{e}$, a proportional controller based on the Lyapunov method is designed according to

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} D/2 & D/2 \\ D/l & -D/l \end{bmatrix}^{-1} \mathbf{M}^{-1}\dot{\mathbf{e}} = \begin{bmatrix} D/2 & D/2 \\ D/l & -D/l \end{bmatrix}^{-1} \mathbf{M}^{-1}(-k\mathbf{e}) \tag{14.21}$$

Here, k is the scalar proportional gain, with $k > 0$. The control law is obtained by substituting Equation 14.17 into Equation 14.21, as

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = -k\begin{bmatrix} D/2 & D/2 \\ D/l & -D/l \end{bmatrix}^{-1} \mathbf{M}^{-1}\begin{bmatrix} -s_x(r - r_d) \\ -s_y(c - c_d) \end{bmatrix} \tag{14.22}$$

In Equation 14.22, the pixel coordinates r and c can be directly measured from the current image using the available image-processing software. Therefore, according to Equation 14.22, the desired angular velocities of the two wheels of the mobile robot can be directly computed from the image measurements. Moreover, it is noted that the developed controller guarantees asymptotic stability of the closed-loop system in view of $\dot{\mathbf{e}} = -k\mathbf{e}$.

14.4.1.5 Experimental Results

In this section, the vision-based mobile grasping system is employed to validate the visual servo controller. In the experiment, the mobile robot employs its onboard CCD camera to continuously observe the position of the target object (toothpaste tube with a red cap) shown in Figure 14.13 with the ACTS color blob–tracking software and computes the visual error **e** on the image plane. Next, the IBVS controller determines the desired wheel speeds ω_1 and ω_2 using Equation 14.22 and accordingly commands the low-level controller of the mobile robot to gradually move the robot close to the target object. When the robot is sufficiently close to the object, it grasps the object with assistance of the onboard 5DOF camera. The overall process of the experiment is shown as a still image serial in Figure 14.17.

The trajectory of the visual feature point on the image plane is shown in Figure 14.18. In Figure 14.18, the initial position of the visual feature is close to the top right corner of the image and then moves directly toward the desired position at the bottom left corner.

The position and heading history of the robot in the entire process is shown in Figure 14.19, the pixel errors on the image plane are shown in Figure 14.20, and the control inputs are given in Figure 14.21.

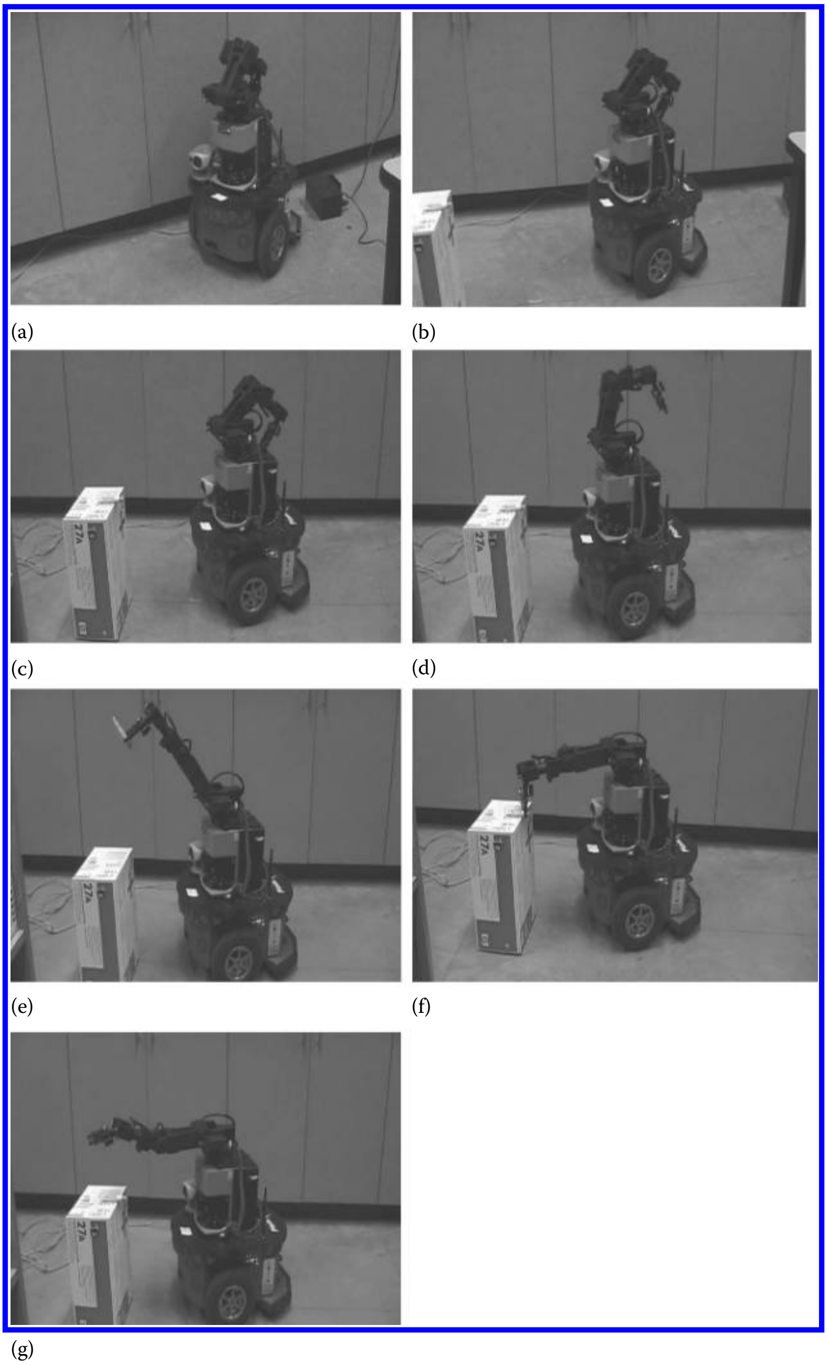


FIGURE 14.17
Vision-based autonomous grasping using a wheeled mobile robot and the developed IBVS controller.

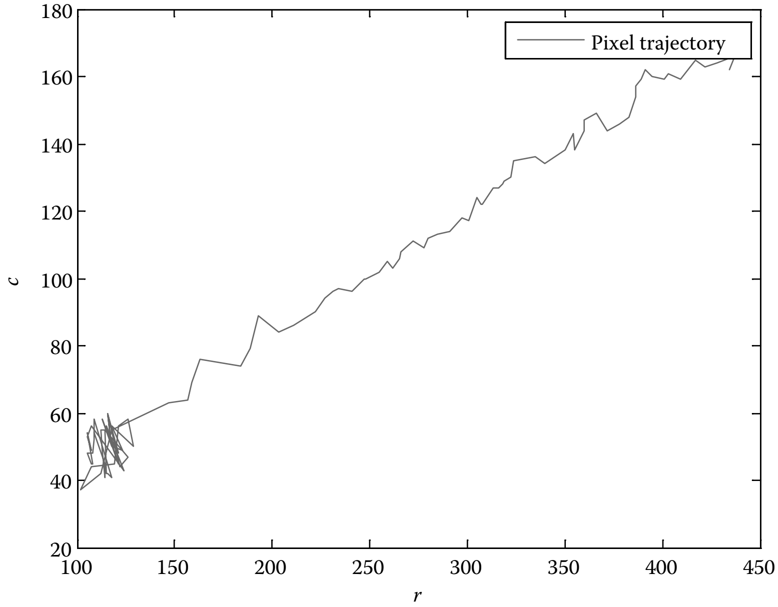


FIGURE 14.18
Trajectory of the visual feature point (object) on the image plane when the robot approaches the object to grasp it.

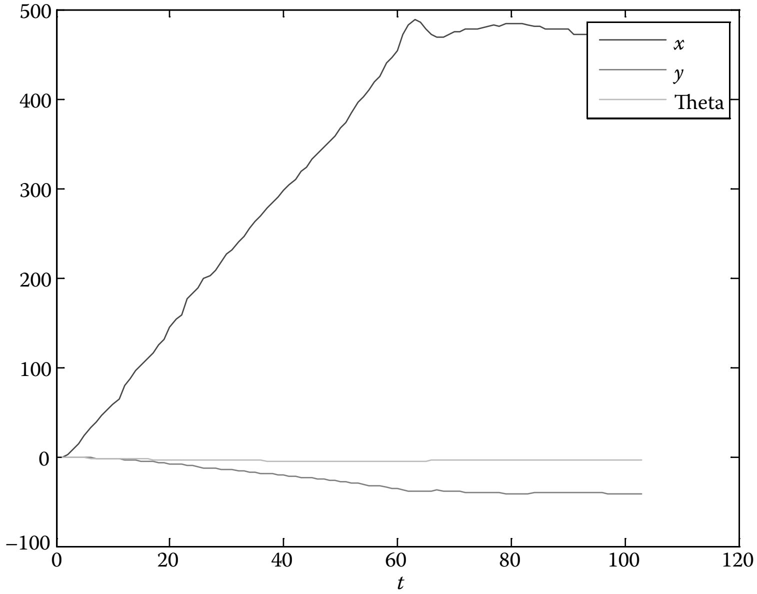


FIGURE 14.19
Trajectory (position and heading) of the mobile robot in the physical environment for the mobile manipulation task.

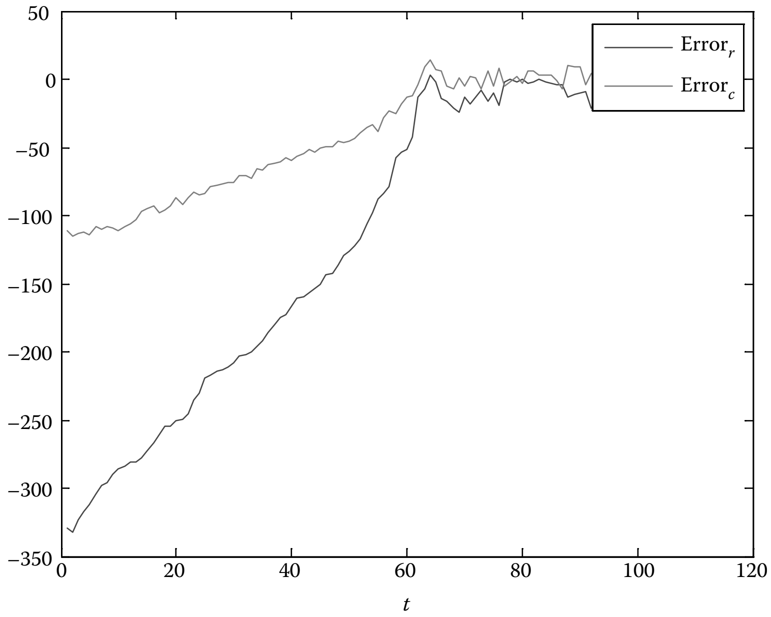


FIGURE 14.20
Visual errors on the image plane when the robot approaches the object to grasp it.

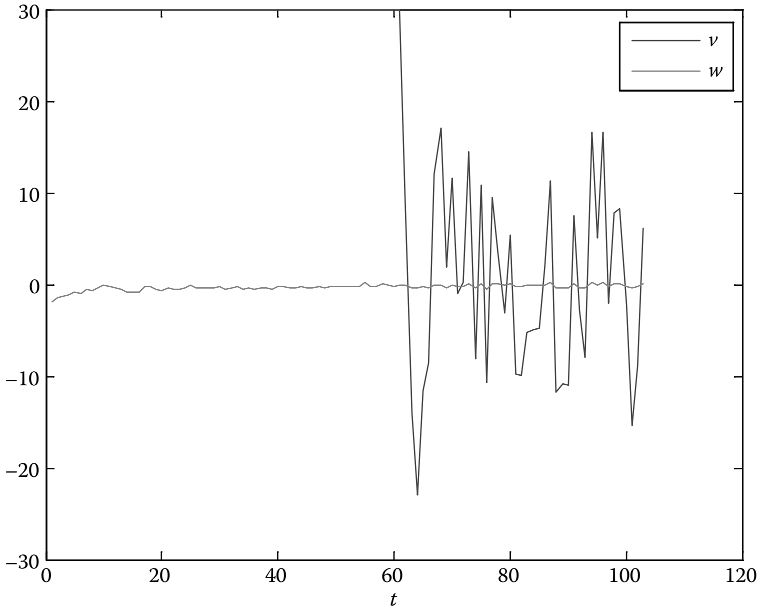


FIGURE 14.21
Control inputs of the plant under the new visual servo controller.

From Figures 14.17 through 14.21, it is clear that the developed IBVS controller is able to effectively drive a wheeled mobile robot toward a target object until its CCD camera observes the desired visual features.

14.4.2 Hybrid Controller for Robust Visual Servoing

14.4.2.1 Control Scheme

Although the experimental results in the previous section validate the proportional controller for visual servoing of a wheeled mobile robot, some shortcomings are still present in the approach. Most importantly, this controller cannot guarantee the retention of visual features within the image plane. On one hand, in order to increase the speed of response and to reduce steady-state errors, the controller gain has to be sufficiently high. On the other hand, when the controller gain is too big, the control inputs (v and ω) increase correspondingly, and as a result, the visual features can quickly move out of the image plane, which leads to failure of the controller. In view of such shortcomings, it is necessary to improve the robustness of the controller developed in the previous section. In the present section, a hybrid switching controller is developed to eliminate the main shortcomings of the previous controller [6]. The new control scheme is schematically presented in Figure 14.22.

In Figure 14.22, there are two control loops: the IBVS controller developed in the previous section and a new Q learning–based controller. The IBVS controller will drive the mobile robot toward the target object, and the Q-learning controller will observe the visual features on the image plane and select an optimal action (an appropriate rotational or translational movement) so that the visual features are pushed from the image edge to the center. In addition, the Q-learning controller is able to continuously learn and improve its action–selection policy online using its machine-learning algorithm.

There is a rule-based arbitrator in the control system as indicated in Figure 14.22, which autonomously switches between the outputs of the controllers so that the overall hybrid control system achieves a tradeoff between its robustness and accuracy.

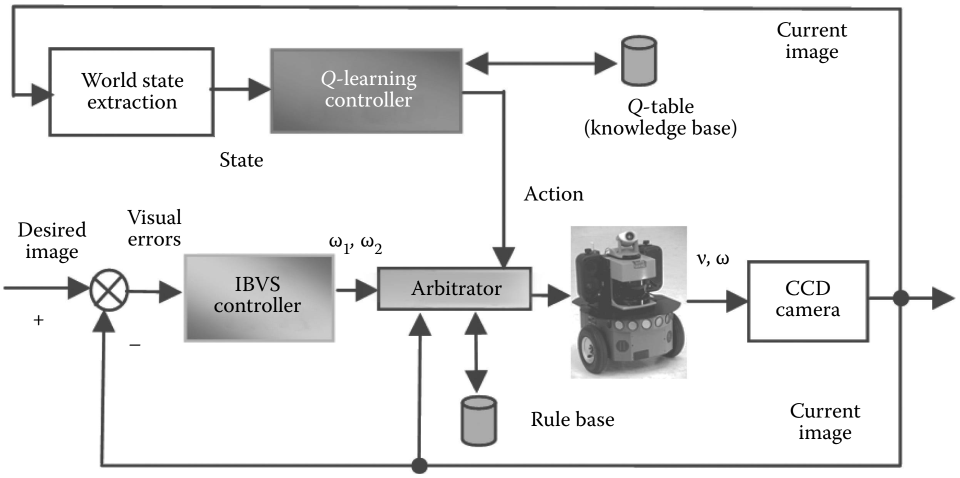


FIGURE 14.22
Hybrid control scheme for robust visual servoing.

The *Q*-learning controller can be trained offline to improve its performance. Once it is trained, the *Q*-learning controller can quickly select correct actions in real time (usually in less than 10 msec in the experiments).

14.4.2.2 Q-Learning Controller

The *Q*-learning controller indicated in Figure 14.22 is a customized controller, which will keep the visual features within the image plane. It is based on the machine learning approach *Q* learning. The main advantage of the *Q*-learning controller is that it is able to autonomously learn the action–selection policy of the mobile robot and improve the controller performance continuously so that the visual features remain in the field of view of the CCD camera. Due to integration of the *Q*-learning controller with the IBVS controller developed in the previous section, the robustness of the resulting hybrid controller is improved.

14.4.2.2.1 States, Actions, and Rewards

In this section, a *Q* learning–based discrete event controller is developed to improve the robustness of a vision-based mobile manipulation system. In particular, the controller will continuously grab images from the CCD camera, compute the current world state based on the positions of the visual features in the current image, and will employ the *Q*-learning algorithm to learn and select an optimal action for the robot under the current state. This optimal action (a command for desired translational or rotational motion) will be sent to the low-level controller of the mobile robot so that the visual features move toward a desirable area on the image plane. After the robot takes the action, a reward will be computed based on the new positions of the visual features on the image plane, and this reward will be used to update the *Q* value of the action under that world state.

The definition of the world states in the present *Q*-learning controller incorporates a discrete grid world of the image plane as shown in Figure 14.23.

In Figure 14.23, the 640 × 480 CCD image plane is divided into an 8 × 6 discrete grid world in which each cell of the grid has a length of 80 pixels. When an image is grabbed

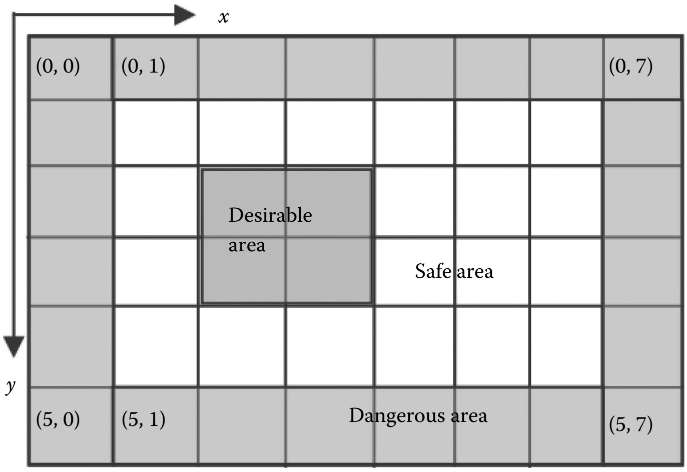


FIGURE 14.23
Discrete grid world defined on the 640 × 480 CCD image plane.

from the CCD camera, the position of the visual feature point on the image plane can be easily converted into the coordinates in the grid world as follows:

$$\begin{aligned} x &= \mathrm{INT}(r/80) \\ y &= \mathrm{INT}(c/80) \end{aligned} \tag{14.23}$$

Here $r = 0,1,\cdots 639$ and $c = 0,1,\cdots 479$ are the pixel coordinates of the visual feature points on the image plane, $x = 0,1,\cdots 7$ and $y = 0,1,\cdots 5$ are the corresponding grid coordinates in the grid world, and INT() is a function that converts a floating point number into an integer by discarding its decimal portion.

The world state in the *Q*-learning controller is made up of the grid coordinates of the visual feature and the current depth from the robot to the target object, as given by

$$s = (x, y, d) \tag{14.24}$$

Here, d is a discrete index value of the current depth, which is computed according to

$$d = \begin{cases} 0, & \text{if depth} < 40\,\text{cm} \\ 1, & \text{if } 40\,\text{cm} < \text{depth} <= 90\,\text{cm} \\ 2, & \text{if } 90\,\text{cm} < \text{depth} <= 140\,\text{cm} \\ 3, & \text{if depth} > 140\,\text{cm} \end{cases} \tag{14.25}$$

Under each state, it is assumed that the mobile robot is able to select one of the following four actions:

- Action #1: Move forward 5 cm
- Action #2: Move backward 5 cm
- Action #3: Rotate 5°
- Action #4: Rotate –5°

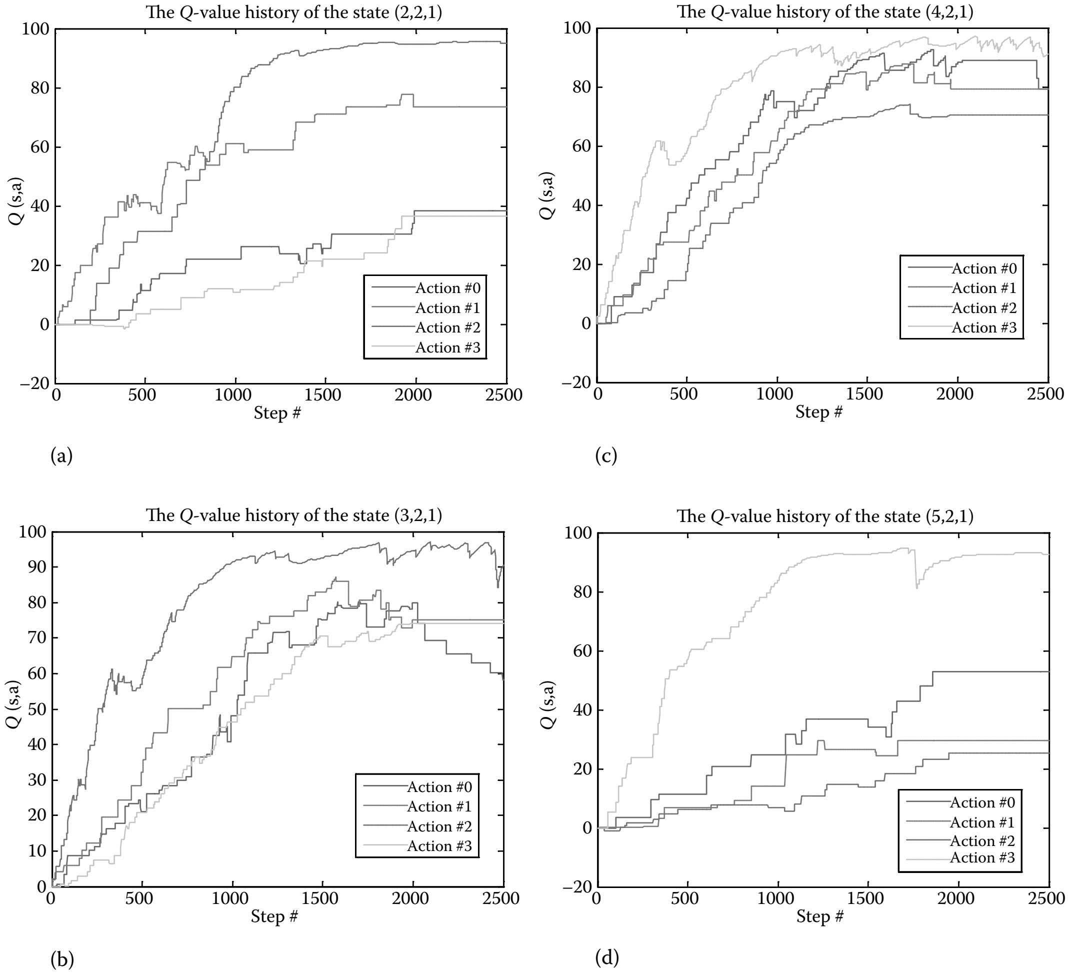
After the robot takes an action, it will receive a reward r from the environment. This reward is computed based on the new position of the visual feature on the image plane, according to

$$r = \begin{cases} +20, & \text{if } (x, y) \in \text{the desirable area} \\ 0, & \text{if } (x, y) \in \text{the safe area} \\ -10, & \text{if } (x, y) \in \text{the dangerous area} \\ -20, & \text{if } (x, y) \text{ is out of the image} \end{cases} \tag{14.26}$$

Here, (x, y) is the new position of the visual feature in the grid world after the robot takes an action, and the definitions of the desirable, safe, and dangerous areas are shown in Figure 14.23. From Equation 14.26, it is clear that the robot is encouraged to select correct actions, which will push the visual features toward the desirable area and away from the dangerous area on the image plane.

14.4.2.2.2 Training Results

The Q-learning controller as presented here was trained with the physical robot camera system described in Figure 14.13 to gain online experience and then learn or update its Q values. In particular, the robot was required to track the position of the target object in Figure 14.13 with the onboard ACTS color blob–tracking software and to take a series of actions based on the Q-learning algorithm so that the position of the target object remained in the desirable area of the image. In this experiment, the Q-learning algorithm was run for 2500 iterations (or steps), and the history of the Q values under four sample states was recorded as presented in Figure 14.24.



(a) (c)

(b) (d)

FIGURE 14.24
History of Q values under different world states when the robot was trained off-line: (a) state (2,2,1), (b) state (3,2,1), (c) state (4,2,1), and (d) state (5,2,1).

From Figure 14.24 it is observed that the Q values converge after 2500 steps of decision making. It means that the Q-learning controller has learned how to select correct actions for each state. This will enable the mobile robot to move or rotate properly so as to keep the visual features in the desirable area on the image. In particular, for the state of (2,2,1), Figure 14.24a shows that action #2 has a higher Q value than the other three actions, so that this action will have a higher probability to be selected by the Q-learning controller when the state (2,2,1) is observed by the robot. Similar conclusions can be drawn from Figure 14.24b through d.

A trained Q-learning controller is, in fact, a behavior-based decision-making unit. It continuously observes world states and probabilistically selects actions for the robot based on the Q values in the Q table. Unlike the traditional behavior-based system described in [7] in which the behavior rule base is fixed and entirely designed by a human designer in advance, the rule base (i.e., the Q table) of Q learning is learned autonomously when the robot interacts with its work environment, and it can be improved online. Therefore, this Q-learning controller is more desirable than the traditional behavior-based approaches. Furthermore, it is also more advantageous than the approaches of potential field, navigation function, or path planning because those approaches do not have a self-learning mechanism and cannot adapt to a changing world and improve their decision-making abilities online.

14.4.2.3 Arbitrator of the Hybrid Control System

In the hybrid control scheme of Figure 14.22, there is an arbitrator that switches on the IBVS controller or the Q-learning controller based on the position of the visual feature in the grid world in Figure 14.23. In particular, it is a rule-based arbitrator that switches the outputs of the controllers according to the following rules:

- Rule #1: If the visual feature P is in the dangerous area, the Q-learning controller is selected.
- Rule #2: If P is in the desirable area, the IBVS controller is selected.
- Rule #3: If P is in the safe area, no switching action is taken.

These rules, through the Q-learning controller, push a visual feature P back to the desirable region if it is located in the dangerous area. Then, the IBVS controller will drive the robot toward the target object.

14.4.3 Experimental Results

Further experiments are carried out to validate the hybrid control scheme developed in this section. In the experiment, the robot shown in Figure 14.13 is moved close to a target object for autonomously grasping it using the feedback information from its CCD camera and the laser distance finder. Unlike the experiment in Figure 14.17, the present experiments will employ the hybrid controller developed in the previous section to complete the mobile grasping task.

14.4.3.1 Hybrid Visual Servoing with a Small Dangerous Area

In this experiment, the robot carries out the mobile grasping task in the presence of a small dangerous area, using the hybrid controller. The dangerous area is given by

$$r < 60 \quad \text{or} \quad r > 550 \quad \text{or} \quad c < 20 \quad \text{or} \quad c > 450 \tag{14.27}$$

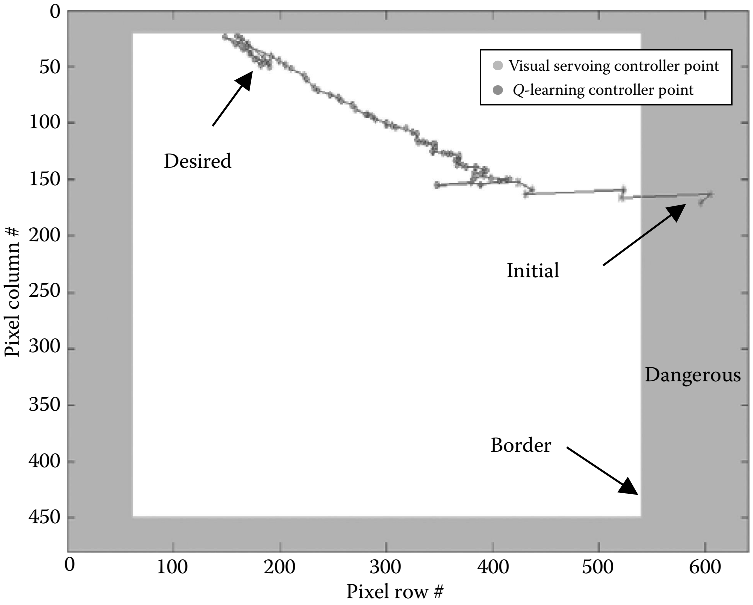


FIGURE 14.25
Trajectory of the visual feature on the 640 × 480 image plane under hybrid control with a small dangerous area.

Where, r and c are the pixel coordinates defined in Equation 14.15.

The trajectory of the visual feature on the image plane is presented in Figure 14.25.

In Figure 14.25, the gray area represents the "dangerous area" on the 640 × 480 image plane, which is defined by Equation 14.27. Because initially the visual feature on the image is at (587,165), which is in the dangerous area, the arbitrator of the hybrid controller selects the Q-learning controller to determine the motion of the mobile robot. From Figure 14.25, it is seen that the Q-learning controller uses just seven steps to bring the position of the visual feature into the "desirable area" on the image plane. After this quick action, the arbitrator selects the IBVS controller to takeover the control of the mobile robot. In the remainder of the control process, the IBVS controller drives the robot to the goal location and orientation until the visual feature reaches its desired pixel position of (180,40) on the image.

Figure 14.25 shows that the developed hybrid controller performs better than the IBVS controller. In particular, when the visual feature is located in the "dangerous area" of the image plane, the hybrid controller is able to quickly adjust the camera pose so that the visual feature will move through a large distance at a fast speed of response so as to enter the "desirable area" on the image. It is usually difficult for the IBVS controller to move the visual feature through a large distance due to potential instability problems.

Figure 14.26 presents the history of the row and column pixel coordinates of the visual feature and the history of controller switching for the experimental result in Figure 14.25.

The performance of the IBVS controller in the hybrid controller is shown in Figure 14.27a through c.

Figure 14.27c indicates that steady-state errors can result when the robot reaches its desired pose. These errors are caused due to the proportional controller given by Equation 14.22. In the present experiments, the steady-state errors are quite small and do not result in a failed grasping.

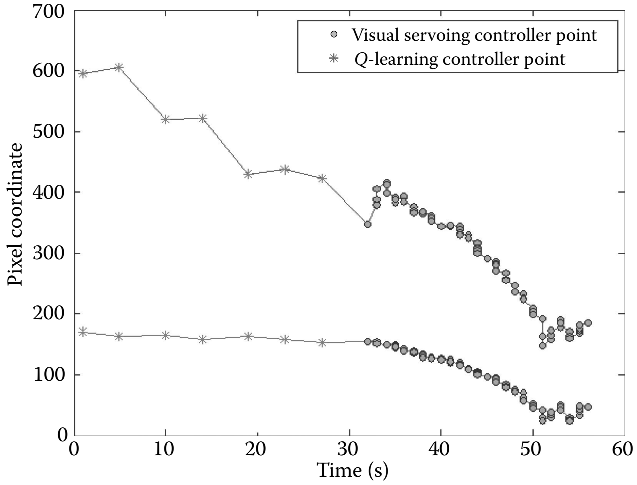


FIGURE 14.26
History of the row and column pixel coordinates of the visual feature when the robot approached the object and grasped it.

14.4.3.2 Hybrid Visual Servoing with a Large Dangerous Area

In this experiment, a large dangerous area is defined on the image plane of the CCD camera, according to

$$r < 60 \quad \text{or} \quad r > 360 \quad \text{or} \quad c < 20 \quad \text{or} \quad c > 260 \tag{14.28}$$

It shows that the hybrid controller is able to drive the robot to its desired position and orientation in a robust manner and make it successfully grasp the target object. Figure 14.28 presents the continuous still images grabbed from the video, showing the overall mobile grasping process.

When the dangerous area is large, the hybrid visual servo controller exhibits very different behavior as shown in Figures 14.29 and 14.30.

As shown in Figures 14.29 and 14.30, because the dangerous area is large, under IBVS control a visual feature can easily enter the dangerous area. Consequently, it is observed that the *Q*-learning controller is activated by the arbitrator very frequently to push back a visual feature into the safe area. Meanwhile, because the visual feature point frequently moves out of the camera's field of view due to the large dangerous area, the robot spends a longer time to complete the task. From both figures, it is clear that the trained *Q*-learning controller has been quite successful in keeping the visual feature within the safe area. Usually, just one step is needed to bring the visual feature away from the dangerous area and then transfer the robot control to the IBVS controller.

In the present work, the *Q*-learning controller makes the robotic system more robust (by keeping a visual feature within the safe area) and the IBVS controller guarantees accurate positioning performance. In this manner, the hybrid controller provides a good tradeoff

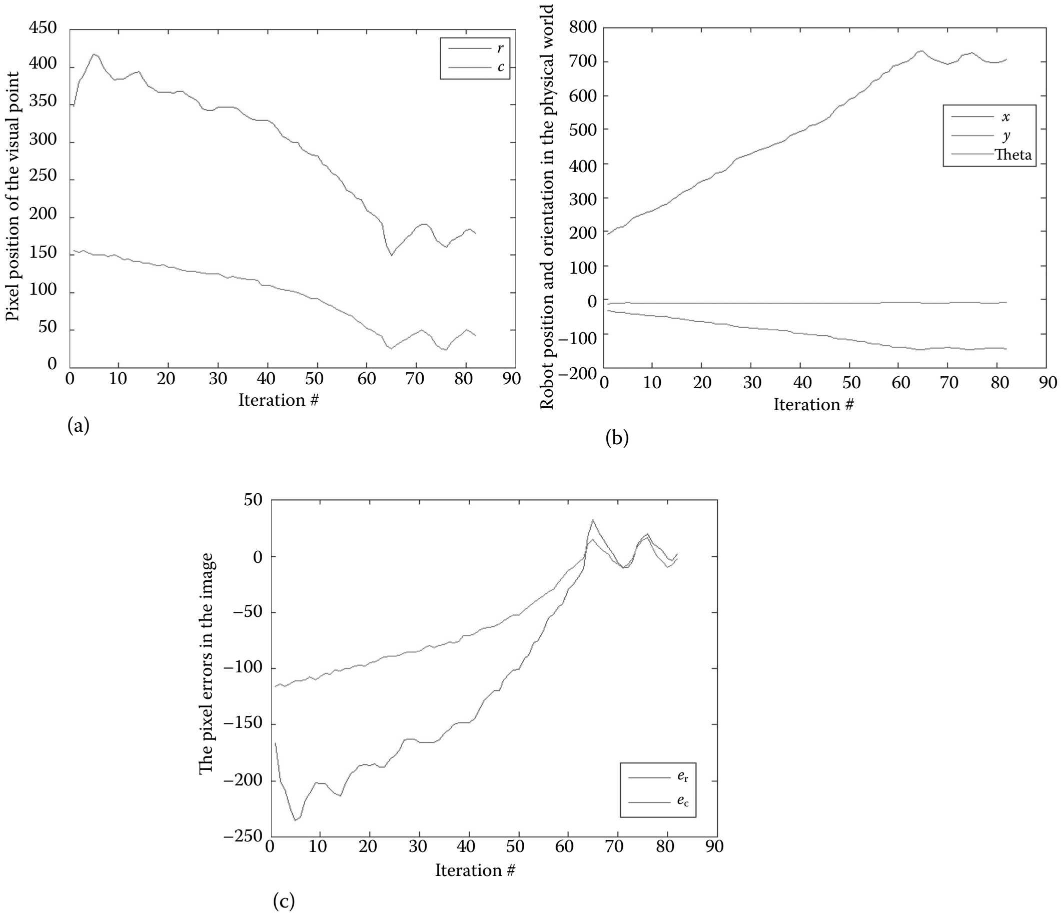


FIGURE 14.27

Performance of the IBVS controller with a small dangerous area when carrying out a mobile manipulation task. (a) The history of the pixel coordinates of the visual feature on the image plane, (b) the trajectory of the mobile robot in the physical environment, and (c) the visual errors on the image plane.

between robustness and accuracy as is clear from Figures 14.29 and 14.30. The performance of the IBVS controller is shown in Figure 14.31a through c.

In Figure 14.31, because initially the visual feature is located in the dangerous area and because the dangerous area is much larger than the one in Figure 14.25, the *Q*-learning controller is activated frequently in the first part of visual servoing to keep the visual feature in the safe area. As a result, it is observed from Figure 14.31 that the response curves oscillate significantly in the first stage (from iteration #0 to iteration #90). After the visual feature is moved away from the dangerous area, the IBVS controller gradually takes control of the mobile robot, and consequently, the response curves become smoother until the visual feature reaches its desired position on the image plane. Again, Figure 14.31 shows that the hybrid visual servo controller developed in this chapter provides a rather "balanced" performance in robustness and accuracy as a result of the integrated IBVS controller and the *Q*-learning controller.

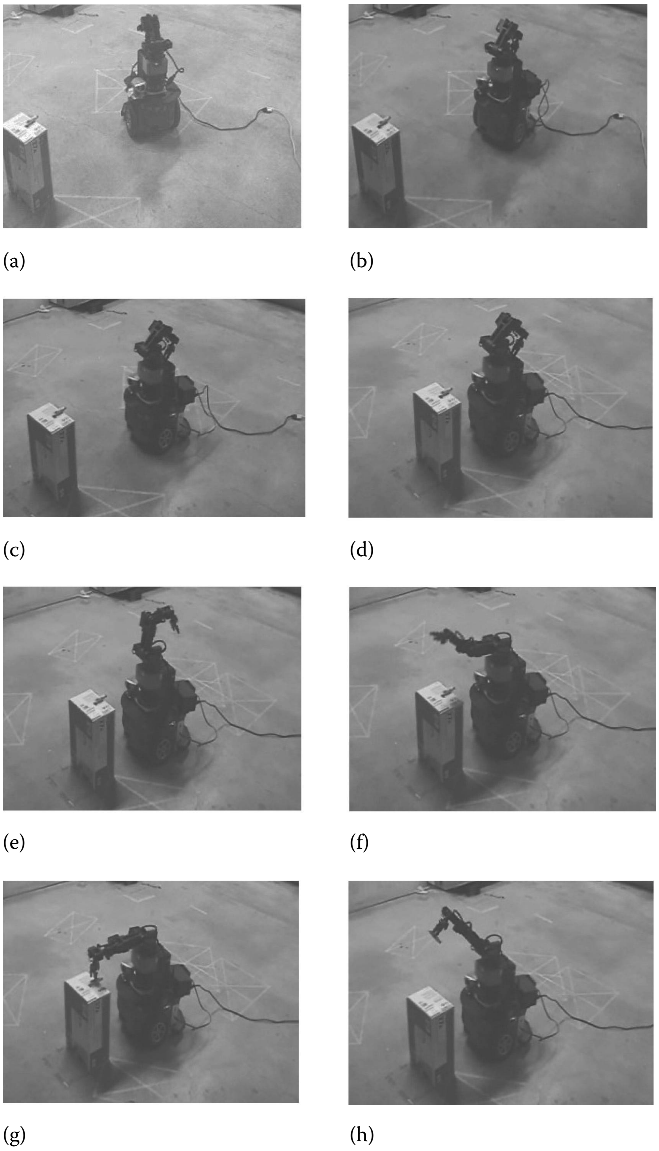

(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 14.28
Mobile manipulation using the hybrid visual servo controller with a large dangerous area. (The battery of the robot is charged continuously through its power cord.)

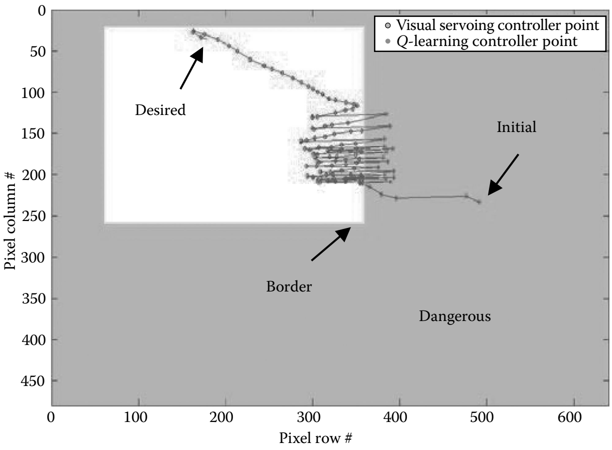


FIGURE 14.29
Trajectory of the visual feature on the 640 × 480 image plane (large dangerous area).

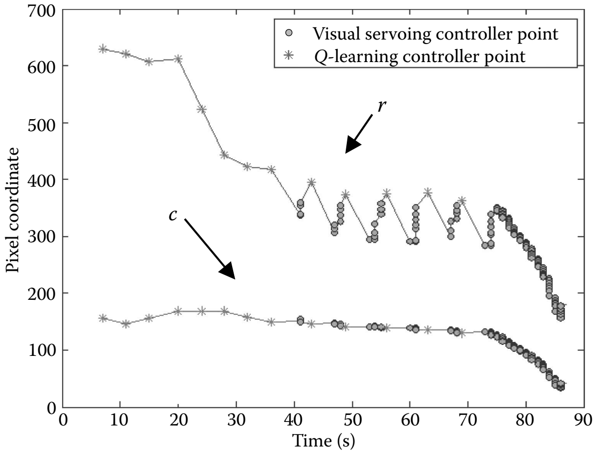
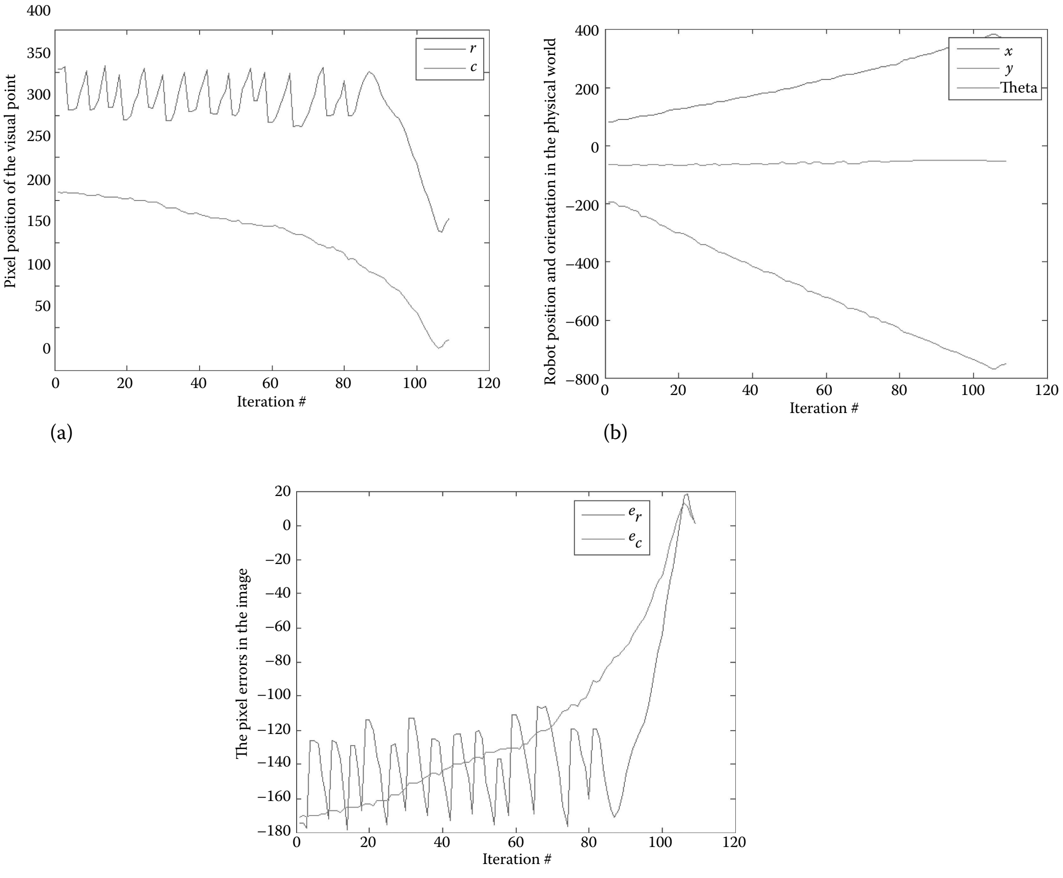


FIGURE 14.30
History of the row and column pixel coordinates of the visual feature.



(a) (b) (c)

FIGURE 14.31
Performance of the IBVS controller when the dangerous area is large. (a) The history of the pixel coordinates of the visual feature, (b) the trajectory of the mobile robot, and (c) the visual errors on the image plane.

14.5 Conclusions

Machine learning is an important area in robotics, particularly when robots are operating in dynamic and unknown environments. Through learning, robots reduce the environmental uncertainties, increase their reliability and improve their performance. In this chapter, one of the popular machine-learning algorithms, reinforcement learning or *Q* learning, was introduced and demonstrated using physical mobile robot systems. Based on the simulation and experimental results, machine learning can help robots cope with dynamic and unknown environments effectively.

References

1. Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Second Edition, Pearson Education, Inc., Upper Saddle River, NJ, 2003.
2. Karray, F. O. and de Silva, C. W., *Soft Computing and Intelligent Systems Design*, Addison Wesley/Pearson, New York, 2004.
3. Wang, Y. and de Silva, C. W., "A machine learning approach to multi-robot coordination," *IFAC International Journal Engineering Applications of Artificial Intelligence (Elsevier)*, Vol. 21, No. 3, pp. 470–484, 2008.
4. Mitchell, T. M., *Machine Learning*, McGraw-Hill Companies, Inc., New York, 1997.
5. Spong, M. W., Hutchinson, S., and Vidyasagar, M., *Robot Modeling and Control*, John Wiley & Sons, Inc., Hoboken, NJ, 2006.
6. Wang, Y., Lang, H., and de Silva, C. W., "A hybrid visual servoing controller for robust manipulation using mobile robots," *IEEE/ASME Transactions on Mechatronics*, Vol. 15, No. 5, pp. 757–769, 2010.
7. Arkin, R. C., *Behavior-Based Robotics*, The MIT Press, Cambridge, MA, 1998.

15

Neuromechatronics with In Vitro *Microelectrode Arrays*

Dulini Mendis, Steven Petrou, and Saman K. Halgamuge

CONTENTS

15.1 Introduction 568
  15.1.1 Evolution of Mechatronics 568
  15.1.2 Neuromechatronics 568
  15.1.3 Neuronal Networks 569
15.2 *In Vitro* Microelectrode Arrays (MEAs) 570
  15.2.1 MEAs among Other Neural Recording Techniques 570
    15.2.1.1 Optical Methods 570
  15.2.2 Functionality of MEAs 570
  15.2.3 Strengths and Weaknesses of MEAs 571
  15.2.4 MEA Systems and Software 572
15.3 Dynamics of Microelectrode Array Recordings 573
  15.3.1 Spikes 573
  15.3.2 Bursts 573
  15.3.3 Network Bursts 574
15.4 Detection of Network Dynamics 575
  15.4.1 Spike Detection 575
  15.4.2 Spike Sorting 575
  15.4.3 Burst Detection 576
  15.4.4 Network Burst Detection 577
  15.4.5 General Analysis Methods 577
  15.4.6 Identifying Functional Motifs 577
15.5 Embodied Neural Networks 578
  15.5.1 Supervised Learning 579
    15.5.1.1 Animat: A Neurally Controlled Virtual Animal 579
    15.5.1.2 Hybrot: A Neurally Controlled Robot 580
    15.5.1.3 MEART: A Neurally Controlled Robotic Arm 581
    15.5.1.4 Other Attempts at Hybrid Systems 581
  15.5.2 Unsupervised Learning 581
    15.5.2.1 Inducing Plasticity through Training 582
    15.5.2.2 Use of Artificial Neural Networks for the Neural–Robotic Interface 583
15.6 Conclusion 585
References 585

SUMMARY This chapter focuses on recording neural network activity *in vitro* (outside their normal biological system) through microelectrode arrays, an electrophysiological recording method that captures extracellular voltages of a neuronal network and proceeds to explore how *in vitro* neural networks can be combined with robotics to create hybrid systems that are composed of biological brains and artificial bodies.

15.1 Introduction

15.1.1 Evolution of Mechatronics

Mechatronics can be defined as a mixture of technologies and techniques that assist in designing better products. A mechanical system requiring control or/and data manipulation can be designed cheaper, better, and with more flexibility using mechatronics. Over the last 40 years, mechatronics has evolved dramatically, spanning across numerous fields and challenging the borders of its own "parent" disciplines.

The relationship between biology and mechatronics has been beneficial to both disciplines. Mechatronics is widely utilized as an enabling technology in medicine in the creation of instrumentation and devices, such as surgical devices, endoscopy, rehabilitation, and assistance and micro-array technology. In addition, control engineering, a key component of mechatronics, assists in the understanding of gene regulatory networks and other complex control systems in biology. "Robot-in-the-loop" systems also play a key role in complex biological experiments.

Mechatronics has also benefitted from biological insights in the development of bio-inspired robots (biomimetic robotic systems, such as humanoids and animaloids), designs that mimic the body's musculoskeletal design, bio-inspired sensory-motor coordination, control algorithms that use principles of biological movement, and bio-inspired learning in mechatronic systems.

15.1.2 Neuromechatronics

In the vast and expanding field of biomechatronics, neuromechatronics can be broadly defined as the integration of mechatronics with neural systems. Generally this involves a closed loop system in which neural systems receive input from mechatronic devices and/or control a mechatronic component. Hence, the field of neuromechatronics encompasses biological signal acquisition, signal manipulation, and creation of the interface between biological and mechatronic systems.

The human nervous system consists of the central nervous system (CNS), containing the brain and the spinal cord, which acts as the information processing hub, and the peripheral nervous system (PNS), which holds the input and output "terminals." The PNS contains afferent neurons, which carry information from sensory organs to the CNS, as well as efferent neurons that carry out information from the CNS to muscles or glands to initiate action. In the case of a neuromechatronic system, sensory organs can be replaced by artificial devices that generate stimuli for the hybrid system, and muscles can be replaced by artificial actuators, such as robotic arms.

This chapter focuses on this aspect of neuromechatronics. However, note that this is different from the concept of mechatronic devices replacing the functionality of the nervous system as in neural prostheses. In such instances, a mechatronic device is controlling a non-neural biological component by mimicking the functionality of a neural component, or it is serving as the bridge between neural and non-neural components (e.g., using signals from motor neurons in the arm to control a robotic hand).

It is clear that neuromechatronic systems can be constructed at various scales of the neural system. The integration of electroencephalographic (EEG) or electromyographic (EMG) signals to control prosthetic limbs involves neural activity recorded at a macroscopic level while innervating a single nerve using artificial stimulations to observe resulting motor actions involves a lower-level system. In this chapter, the focus will also be at the level of neuronal networks.

15.1.3 Neuronal Networks

On average, a neuron from a mature human nervous system is connected to 10,000 other neurons. Signals are passed from neuron to neuron at electrical or chemical connections called synapses. The three primary components of a typical neuron are the soma or cell body, the dendritic arbor that serves as the input structure, and a single axon that serves as the output. A synaptic connection is normally made between the "bouton" of an axon of one neuron and a dendritic "spine" of downstream neurons. Neuronal activity is measured at the most basic level via the action potentials that they generate, which are referred to as spikes.

Although high-level tasks, such as cognition, speech, or motor control involve the integrated function of multiple brain regions, it is intriguing to assess the ability of networks of neurons to accomplish simpler tasks. Networks of neurons initially self-assemble to perform a predetermined task, or they may adjust themselves to "learn" new tasks after initial formation because of inherent structural and functional "plasticity."

Plasticity enables neurons and the networks in which they reside to change their functions to reflect changes in implicit or explicit processes. Synaptic plasticity is, therefore, the changes in the properties of the synaptic connections between neurons, which, in turn, affects neural network function. Learning can be considered to be a more stable state of plasticity in which neurons alter their activity corresponding to a relatively constant external change. Plasticity and learning are of high importance because they build the foundation for the concept of memory.

Plasticity and learning have been extensively studied by a range of electrical recording approaches. The extracellular manifestations of synapses are recorded as "field potentials," and those of action potentials recorded as "units." Intracellular recordings either employ sharp microelectrodes that penetrate the cell membrane or blunt patch clamp electrodes that press onto the surface and access the cell by rupturing the membrane. These methods can record neuronal membrane potentials or, by employing feedback, can measure membrane currents. Learning emerges more apparently at the network level rather than the single neuron level at which plasticity can be demonstrated, but its relationship to learning is not clear. Such complex tasks require knowledge of the state of the network and input and output activity across several brain regions emphasizing the importance of network state analysis.

Network-level analysis also provides insight into the computational capabilities of biological networks as well as how they achieve computational complexity with minimal power requirements, highly relevant to fields such as neuromorphic engineering. Moreover, it allows probing into the small world connections that may be carrying out elementary functions and operating independently in small clusters of neurons as opposed to high-level connections between functional regions of the brain that coordinate to perform higher level cognitive tasks.

The timing of action potentials has been extensively studied as the means of encoding information as opposed to firing rates but mostly at the level of single neurons. Therefore, investigating time-wise correlations between multiple neurons as well as time-dependent higher order network dynamics can also yield information about how these networks function.

15.2 *In Vitro* Microelectrode Arrays (MEAs)

15.2.1 MEAs among Other Neural Recording Techniques

Studies on network-level sensory stimulation and neuronal responses have been prominently addressed through technologies such as electroencephalography (EEG), magnetoencephalography (MEG), and functional magnetic resonance imaging (fMRI). These methods enable noninvasive time and frequency analysis of neural signals but at relatively low resolution. On the other end, techniques such as single unit recordings and the patch clamp methods provide a higher resolution, targeting single cells and ion channels, respectively, but in spatially restricted regions. MEAs are placed at an intermediary level in this spatial resolution scale in which electrical signals from neuronal networks of thousands of neurons are recorded in parallel, thus providing insight into neuronal interconnections and network activity.

15.2.1.1 Optical Methods

Optical imaging techniques that capture fluorescent chemical indicators or genetically encoded proteins are another set of technologies that can address high spatial and temporal scales in single or small networks of neurons. Two-photon calcium imaging has the ability of recording tens or hundreds of neurons simultaneously both *in vivo* and *in vitro* by capturing the changes in Ca^{2+} levels of cells using chemical indicators or genetically encoded calcium indicators (GECI). Compared to MEAs, calcium imaging provides higher spatial precision but lower temporal resolution. However, the signal-to-noise ratio, kinetics, and response speed of indicators are continually being enhanced, especially in GECIs such as GCaMP [1,2].

Optogenetics is another key area in which single neurons or ensembles of neurons are optically stimulated so that neurons with genetically encoded light-sensitive proteins are activated. The response of neurons is captured through indicators sensitive to ionic concentrations or membrane voltages. Optogenetics have at present advanced to a state in which simultaneous stimulation and recordings can be carried out for a large number of neurons. Their individual membrane potentials, synaptic potentials, spiking, and bursting can be captured at micrometer spatial resolution and millisecond temporal resolution [3].

With its rapid development in spatiotemporal precision, optics has emerged as a key technology in exploring neuronal network dynamics and learning [4,5], mainly in *in vivo* studies. However, it has not yet been fully employed in learning and plasticity studies *in vitro*. It would especially be interesting to investigate what optics could uncover when used in hybrid closed-loop systems comprising *in vitro* networks.

15.2.2 Functionality of MEAs

In vitro microelectrode arrays (also referred to as multielectrode arrays) are an extracellular neural recording technology that has shown promise in detecting phenotype changes

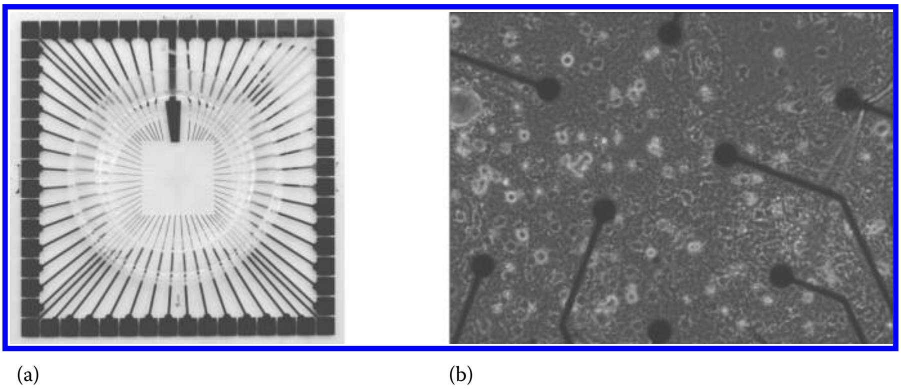

(a) (b)

FIGURE 15.1
(a) Single-well MEA. (Multichannel Systems [11].) (b) Cultured neural networks (CNNs) from mouse frontal cortical neurons growing on a standard MEA.

in neuronal networks in the presence of stimulations and pharmacological agents [6–9]. They can be used with any electrogenic tissue (i.e., brain/cardiac slices, dissociated cultures, retina, stem cells) and enable studies of normal and induced physiological functions that mimic *in vivo* damage [10].

Standard MEAs consists of an array of electrodes integrated into a transparent substrate over which a dish containing a neuronal culture is placed (Figure 15.1) [11]. Different types of MEAs can also be used for tissue slices or retinal recordings and can record field potentials or units in this mode. Neurons and their interconnections are more or less preserved in acute tissue slices, and in dissociated cultures, interconnections are formed in the MEA after plating, allowing observations of the growth of neuronal networks but without the precise structure of *in vivo* formed networks.

Neuronal action potentials or "firing" is accompanied by extracellular current flow and a corresponding voltage gradient in the extracellular fluid, which varies in temporal and spatial domains. MEAs capture these extracellular voltages by simultaneously recording from multiple electrodes that are spatially distributed over the neuronal network. MEAs may contain a single well in which the entire neuronal network grows as a single cluster or multiple wells in which a separate network is confined by each well, allowing high throughput experiments in which multiple conditions can be tested in parallel.

15.2.3 Strengths and Weaknesses of MEAs

The higher resolution and long-term extracellular recording capabilities that MEAs offer make it a useful technology for assessing spatiotemporal patterns in neuronal networks. Unlike their *in vivo* counterparts, *in vitro* recordings provide greater versatility in changing experimental conditions and recording times. Single recording events can extend over several hours and conditions, and cultures can be retained for at least a month, providing useful insights into network changes as they mature and interconnections develop.

Intracellular recordings, such as patch clamp and sharp electrodes, offer high accuracy in recording electrophysiological activity of neurons. Additionally, they are capable of capturing a wide range of voltage dynamics expressed by cells whereas MEAs used with cultured neuronal networks only capture the binary event of action potential firing. However, the former methods limit observation times due to mechanical and biophysical instabilities and require bulky micromanipulators for steering electrodes into cells [12].

The ability of scaling MEAs to provide higher resolution [13], ability to combine electrophysiological and optical recording techniques, and the capability of creating separated cell networks and custom recording structures [14] enhances their usability in functional network studies.

Even though they are initially isolated neuronal networks, *in vitro* tissue or cultures can readily be used in assessing the initial variability of neuronal activity for constant stimuli as well as the reduction in variability with prolonged training, which enables them to be utilized in learning and plasticity studies as well as closed-loop robotic systems. Moreover, understanding spontaneous and stimulated neuronal behavior increases the ability to create better interfaces between neurological and artificial systems, which is informative for the progress of neural prosthetics.

On the other hand, MEAs fail to capture neurons that are silent or are firing at extremely low frequencies. These neurons may hold information about sparse coding in networks in which only a small set of neurons correspond to a particular task. Hence, the loss of such information is of a significant impact. Moreover, MEAs cannot capture subthreshold voltages or fully distinguish between the sources of recorded potentials, which impairs in-depth investigations into synaptic pathways and signaling [12].

15.2.4 MEA Systems and Software

MEAs are microfabricated and integrated into glass or silicon substrates with the number of electrodes ranging from 32 in sparse arrays to more than 4000 in high-density MEAs. Recording frequencies as high as 50 kHz has been enabled as at present. Commercial systems include signal amplifiers, temperature control systems, and parallel recording capabilities for multiple MEAs as well as real-time stimulation and feedback.

Over the years, there have been a number of software tools made available for the analysis of MEA recordings, both commercial and open source. Many of the commercial MEA system vendors offer packages for online as well as offline data analysis. Figure 15.2 displays the main commercial MEA systems and associated software.

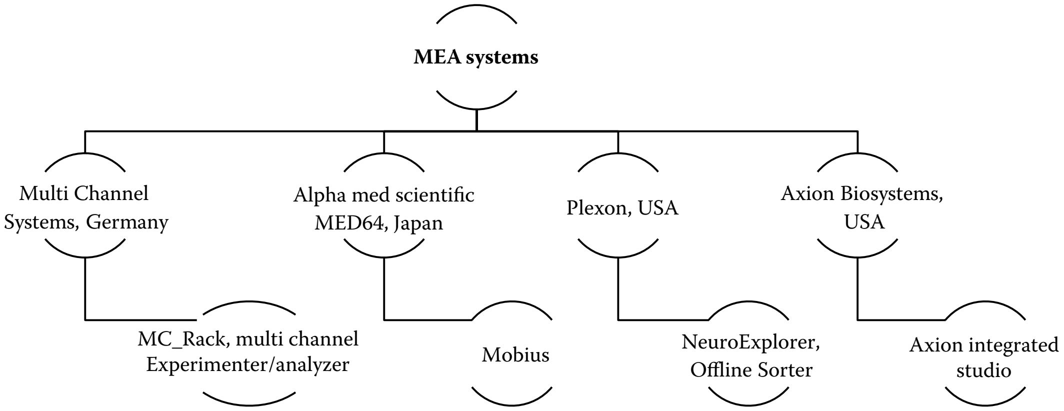

MEA systems
Multi Channel Systems, Germany
Alpha med scientific MED64, Japan
Plexon, USA
Axion Biosystems, USA
MC_Rack, multi channel Experimenter/analyzer
Mobius
NeuroExplorer, Offline Sorter
Axion integrated studio

FIGURE 15.2
Commercial MEA systems and associated software.

Most of these packages offer recording, playback, and basic analysis tools, such as filtering, spike detection, generation of raster plots, etc. NeuroExplorer, which is created by Plexon, offers additional tools for analyzing correlations, spectral densities, and stimulation responses. Plexon also produces the Offline Spike Sorter software, which sorts spikes depending on spike shape using a range of clustering methods.

On the other hand, the research community has also introduced several software packages to assist with MEA data.

1. MEATools [15] includes spike analysis and local field potential (LFP) analysis.
2. FIND [16] offers spike detection and metric generation.
3. MeaBench [17] is a Linux-based tool set that includes real-time processing and visualizing, online spike detection and offline burst detection.
4. SpyCode [18] offers spike detection, spike analysis, burst detection, cross-correlation analysis and avalanche analysis [19].

15.3 Dynamics of Microelectrode Array Recordings

In order to describe the current state of the neuronal network and to relate it to afferent commands, electrophysiological features of the network have to be analyzed. The raw recordings from MEA electrodes provide analog waveforms representing extracellular voltages at different spatial points of the network. Components of the analog signal, such as the local field potentials (LFP), the slowly developing signals thought to arise from the concerted action of multiple activated synapses, can be used to characterize the state of the network as LFPs represent the input to the network. However, the timing and rate of individual action potentials have the potential to carry higher information content than LFPs. Hence, voltage waveforms are converted into point processes in most MEA applications.

15.3.1 Spikes

The fundamental element in MEA recordings is a "spike" or "unit," which is the time stamp of the extracellular voltage that rises above noise and corresponds to action potentials of neurons. Spikes recorded via a single channel (electrode) may originate from a single or multiple neurons depending on the density of the culture and the spacing of electrodes. Spike trains of single electrodes or single neurons are treated as stochastic point processes. The time interval between two spikes is called the interspike interval (ISI), and the firing rate is defined as the number of spikes in a given time interval. Units or spikes represent the output process of a neuron, and together with the LFPs a full input–output relationship can emerge.

15.3.2 Bursts

Whenever, there is a fast progression of spikes in a single channel, these spikes are often found occurring in "bursts" (Figure 15.3). The duration of a burst is the sum of the ISIs of the spikes included in it. An interburst interval (IBI) is defined as the interval between the end of one burst and the start of the following burst.

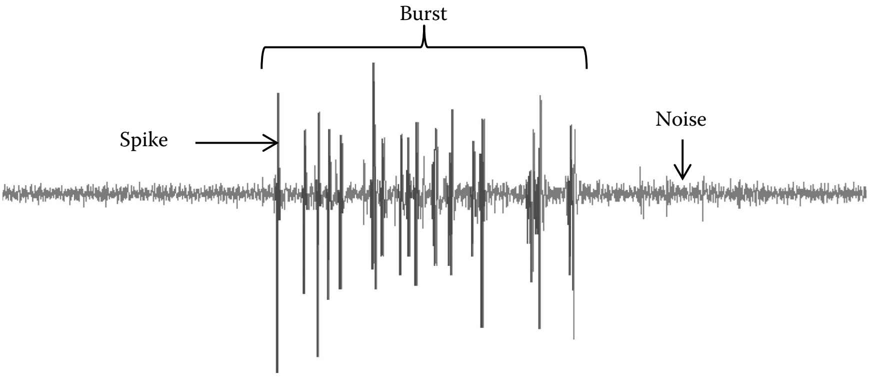


FIGURE 15.3
Time varying extracellular voltage of a single channel.

15.3.3 Network Bursts

If bursts occur across all channels, this phenomenon is referred to as a network burst [20–22] (Figure 15.4). An alternate definition is that network bursts are time periods in which the array-wide firing rate increases.

Array-wide bursting is a dominant feature of *in vitro* recordings unlike in other neural recordings in which bursting is considered to be surprise events [23]. Neuronal cultures are found to exhibit spontaneous bursting starting from 4–7 days *in vitro* as small local bursts that involve a few channels and develop into global network bursts in which almost all of the channels are included as the culture matures.

This synchronized bursting phenomenon has been studied in several labs across the globe [22,24–27]. Network bursts display a wide range of durations, interval distributions,

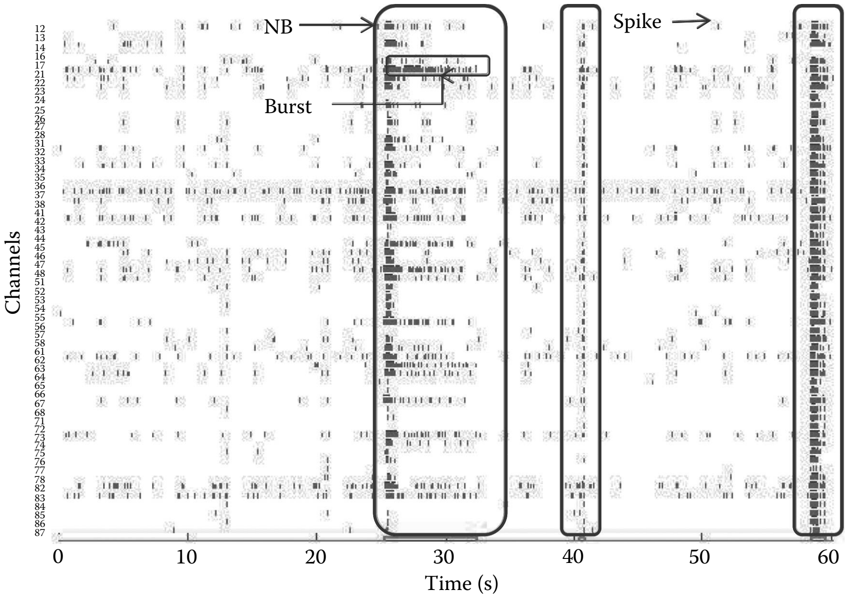


FIGURE 15.4
Raster plot of a 60-channel MEA recording of a mouse cortical culture.

and number of participating channels as well as other dynamics, such as long tails and closely spaced clusters of network bursts (super bursts) [24,27]. The maturity of the culture, the culturing protocols, and pharmacological manipulations all contribute to the diversity of burst profiles in MEA recordings.

Network bursts have been used to identify hidden spatiotemporal structures within neurons using the burst propagation patterns across channels/neurons within network bursts [20,28]. Different propagation patterns are thought of as being related to different functional neuron clusters [29].

15.4 Detection of Network Dynamics

For integrating neuronal networks with artificial systems, data extraction algorithms need to operate as close to real time as possible. However, most of the detection methods proposed for identifying MEA network elements have been designed for post hoc analysis. On the other hand, post hoc methods can be effectively utilized for assessing stimulus selectivity and variations in neuronal responses with time.

15.4.1 Spike Detection

Spike detection is the fundamental step in analyzing MEA data streams and can be performed in real time using most commercial software. A high-pass or band-pass filtering phase is necessary prior to spike detection so that the lower frequencies, including LFPs, are blocked.

Conventional methods depend on thresholding to detect spikes, and a spike is identified if its amplitude rises beyond a multiple of the average noise. Because action potentials are biphasic, conventional thresholding methods only consider one phase of the waveform for detecting the position of spikes. SpyCode offers this conventional method as well as a novel method termed precision timing spike detection (PTSD) [30]. PTSD has the ability to automatically compute the average noise as well consider both phases of the waveform. However, SpyCode only offers post hoc analysis. MeaBench offers online spike detection based on thresholding. In order to provide robust performance amidst drifting noise, it calculates noise as a time-varying signal as well.

15.4.2 Spike Sorting

Without spike sorting, the units or spikes detected cannot be attributed to a single neuron and are referred to as multi-unit recordings. Spike sorting is used to separate spikes detected via a single electrode into single units that can be confidently attributed to the same neuron. This is important because certain functions may be driven by a single neuron or spread across multiple neurons, and this knowledge may be critical for brain–computer interfacing. A range of methods from those based on simple spike parameters, such as amplitude and gradients and independent component analysis (ICA) to wavelet analysis, have been utilized in the attempts to cluster spikes [31,32]. Real-time spike sorting, which is of crucial importance when employing *in vitro* cultures for closed-loop experiments was addressed by Franke et al. [33].

15.4.3 Burst Detection

Early burst detection methods as well as those included in commercial software rely on user-dependent parameters. For instance, in MC_Rack's and NeuroExplorer's burst detection algorithms, several parameters, including minimum number of spikes, minimum interval before and after a burst, and minimum burst duration, are required to be set by the user. User-defined parameters can increase accuracy of results if the experiments are of one particular data type and are indispensable when real-time functionality is required.

If experimental conditions change during a recording session or if a vast number of data sets need to be analyzed, setting parameters for each of these data sets becomes infeasible. Therefore, adaptive burst detection methods were introduced that employ parameters derived from the data itself to adjust the method to suit different data sets. However, this also means that these methods cannot be used in real time.

Almost all adaptive burst detection methods use an ISI threshold to group spikes with ISIs smaller than the threshold into bursts. Wagenaar et al. finds this threshold based on the average firing rate (AFR) [17]. A core ISI threshold is set as min[1/(4 × *AFR*), 100 ms], which is used to find the core section of a burst. A second lenient threshold is used to extend bursts to include adjacent spikes that may be related to the burst (Figure 15.5).

Selinger et al. found that the ISI histogram built using a natural logarithmic scale (ln(ISI)) portrays a bimodal distribution [34] in which the first mode corresponds to intraburst ISIs, and the secondary mode corresponds to the interburst spikes. Hence, a threshold ISI is found at the minimum between the two modes. Pasquale et al. [35] extended this method using a base-10 logarithmic scale instead of the natural logarithm (Figure 15.6) and also added a twofold threshold method similar to Wagenaar et al.

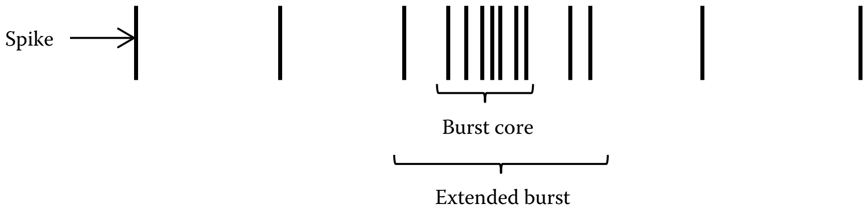


FIGURE 15.5
Burst properties.

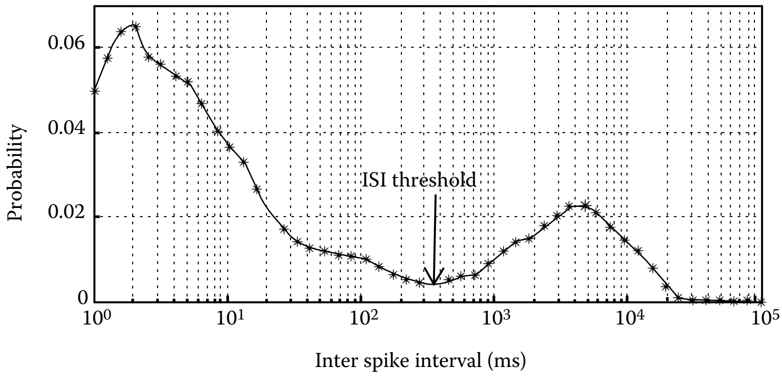


FIGURE 15.6
Bimodal interspike interval histogram of a single channel.

15.4.4 Network Burst Detection

Although single channel bursts provide the foundation for analyzing bursting activity, the more important factor from a network point of view is finding network bursts and propagation patterns. However, most commercial software packages do not include tools for network burst detection. The open source counterparts, on the other hand, include several different network burst–detection methodologies.

One category of network burst detection methods focus on the array-wide increase in activity to identify network burst periods. These methods utilize measures, such as the array-wide spike rate [24,36], product of spike rate, and number of active channels [22], number of neurons fired [28], and array-wide ISI [37].

The other category of network burst detection methods focuses on synchronized bursting when burst detection is required as an intermediary step. MeaBench identifies network bursts using time overlaps of single-channel bursts. Pasquale et al. proposed a more adaptive method based on the log interburst event interval (IBEI) histogram [35]. Given a sufficient recording length, the logarithmic IBEI histogram displays a bimodal distribution similar to that of the ISI histogram. The initial mode of this distribution corresponds to intranetwork burst intervals, and the latter peak corresponds to the internetwork burst intervals. Hence a threshold to group bursts into network bursts is found at the minimum IBEI between the two modes.

15.4.5 General Analysis Methods

Common parameters that are calculated to describe the state of neuronal networks are single values that describe the instantaneous state of the network or a time-averaged history of recordings. In online operations as well as offline analysis, the extracted parameters are primarily related to spikes and bursts. These include the firing rate, burst rate, average number of spikes in bursts, average duration of bursts, ISI, IBI, average spike rate in bursts, etc. [7,9,24].

Bayesian techniques with which distributions of different parameters are analyzed have also been used to describe MEA data so that patterns that are observable in distributions of variables are not averaged out as in the former methods [38].

Correlation and mutual information between channels (or neurons) and correlation between bursts have also been proposed to provide insight into information encoded in networks [18,39,40].

Overall, the MEA data analysis process can be broken into stages as presented in Figure 15.7. However, only spike and burst parameters have been utilized in real-time operations.

15.4.6 Identifying Functional Motifs

Some MEA research is focused on identifying motifs in spikes and bursts, which would provide insight into network states. Motifs are a key factor in closed-loop systems because the aim of training neuronal networks through closed-loop feedback is to reduce variability in motifs in the presence of similar stimuli and link a particular motif with a resultant actuator state.

Repeating motifs of spike patterns were identified by Rolston et al. [41] in MEA recordings, and repeating burst propagation patterns in network bursts were identified by several groups [13,21,42,43]. Eckmann et al. proposed the presence of “leader neurons” that initiate network bursts [44], and Ham et al. introduced the existence of a primary circuit

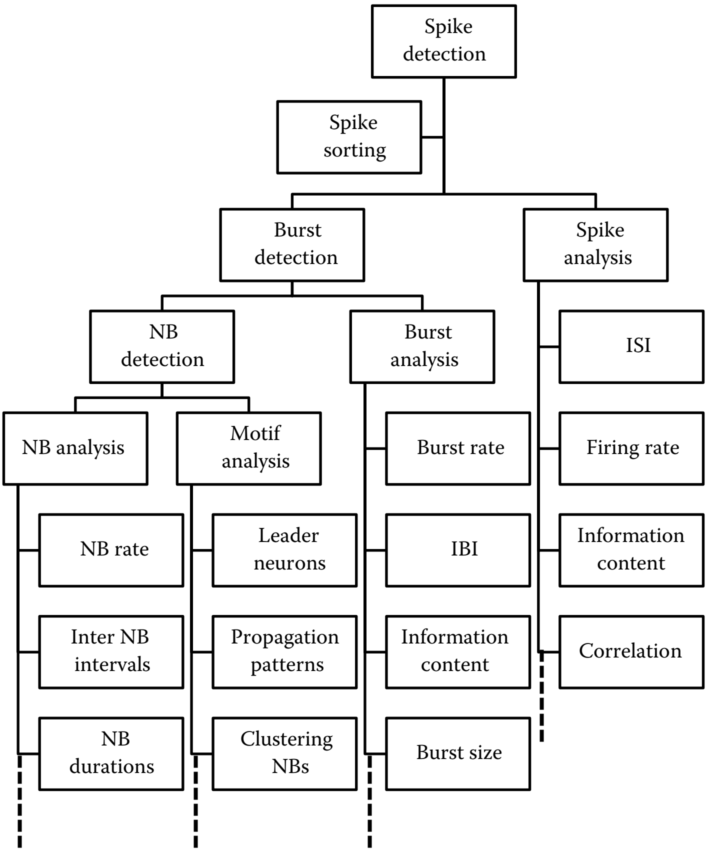

Spike detection
Spike sorting
Burst detection
Spike analysis
NB detection
Burst analysis
ISI
NB analysis
Motif analysis
Burst rate
Firing rate
NB rate
Leader neurons
IBI
Information content
Inter NB intervals
Propagation patterns
Information content
Correlation
NB durations
Clustering NBs
Burst size

FIGURE 15.7
MEA data analysis pipeline.

of neurons that are recruited first in a network burst and assist in propagating the signal to other neurons [45].

Clustering of neurons into functional subunits has also been attempted by several authors [46–48], and clustering is carried out using principle component analysis (PCA) or hierarchical clustering in most cases [46–49]. The features used for clustering ranges from basic parameters, such as firing rate and burst rate to cross-correlation during network bursts and the participation in different burst-propagation patterns.

Bursts themselves have also been clustered depending on features, such as the spike rate within bursts, burst duration, autocorrelation, number of neurons, and Fano factor [50].

15.5 Embodied Neural Networks

The dissociated nature of *in vitro* cultured neuronal networks implies that they are devoid of any external stimuli; hence, they exist in a state of sensory deprivation. The concept of embodying these otherwise isolated networks arose to address the issue of sensory deprivation, so that network behavior will move closer to that of *in vivo* conditions. Hence, it leads to better understanding of network behavior, learning, and neural pathologies. An additional benefit of using *in vitro* methods in studying neural learning is that they allow the continuous analysis of network development during the learning process through electrophysiological recordings as well as through confocal microscopy.

Standalone stimulations provide means of introducing external input into the network. However, embodiment extends past mere stimulations, which involves creating a feedback loop between network response and subsequent stimulations, effectively introducing the ability for the network to interact with its environment. In other words, it creates an input–output mapping for neuronal networks that mimics the link between sensory input and motor output loops of *in vivo* networks.

Therefore, computer simulations and robotics have been employed in embodying these dissociated cultures, which, in other words, is providing an artificial body to an isolated biological brain. It has been hypothesized that networks formed with the influence of external interactions, and hence "experience," resembles *in vivo* networks closer than networks formed in isolation.

15.5.1 Supervised Learning

15.5.1.1 Animat: A Neurally Controlled Virtual Animal

Research on an embodiment initiated more than a decade earlier with the first attempt being the creation of a neurally controlled, computer simulated animal ("animat") by the Potter group [51]. In the "animat" model, a 60-channel MEA was used, and spikes were detected in real time, which were used to construct an activity vector every 200 ms for each channel, which represented the current state. Activity patterns were clustered and each cluster was assigned a direction of motion out of four basic movements; forward, backward, left turn, and right turn. Feedback was given to the network via stimulations for each movement as well as collisions with external objects. The system setup is shown in Figure 15.8.

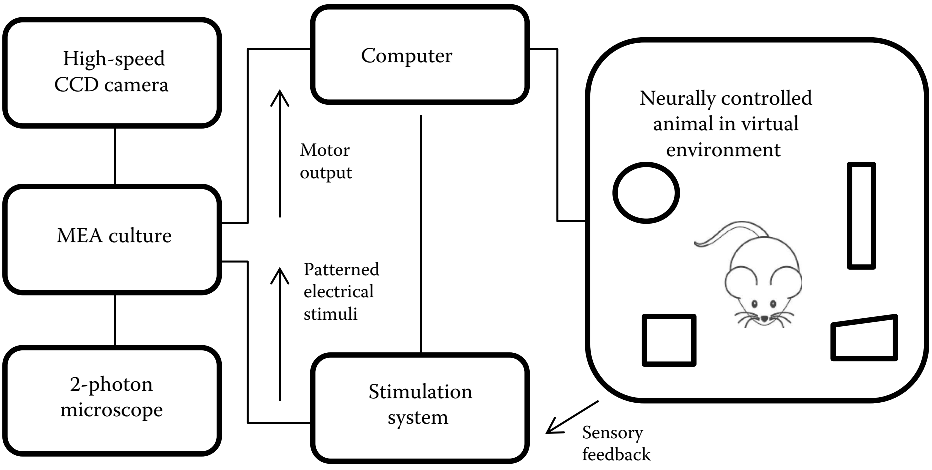


FIGURE 15.8
Animat system setup (modified from [51]). The neuronal culture is microscopically imaged, and the environmental feedback controls the electrical stimulation.

However, a shortcoming in this initial prototype was that movements were initiated due to both stimulated and spontaneous neural activity, and hence, the overall motion of the animat did not follow goal-directed behavior.

15.5.1.2 Hybrot: A Neurally Controlled Robot

Building upon the learnings from the animat model, a neurally controlled robot was later developed, and it was termed "hybrot" (hybrid biological robot) by the same group [52]. The stimuli pattern represented sensory inputs, and the network response, which was measured as the network firing pattern, was converted to motor commands. The aim of hybrot was to study how neuronal networks handled real-time control and adapted itself to the environment.

The hybrot was developed to follow a more natural behavior of "approach and avoidance." In this study, one robot controlled by an *in vitro* neuronal culture was intended to approach another robot but also keep a specific distance from it. The second robot had only restricted movements, and the neurally controlled robot was given the ability to move freely. The distance to target was fed back into the culture in the form of a pair of stimuli at two different electrodes.

The authors found that the neuronal response, as measured by the average firing rate across all electrodes, followed a nonlinear trend dependent on the interstimulus interval in which response to shorter intervals was higher as opposed to longer intervals (Figure 15.9).

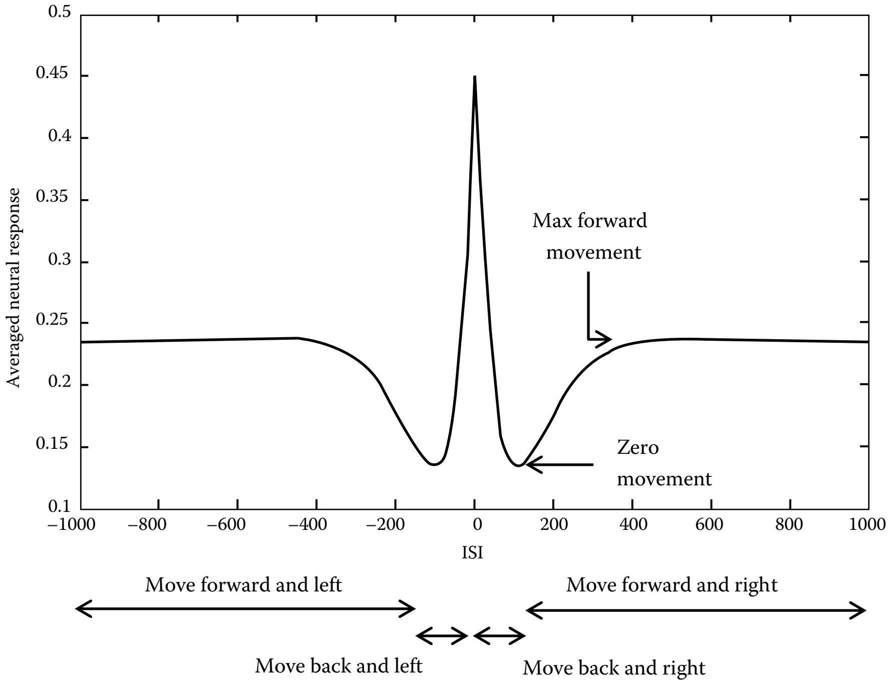


FIGURE 15.9
Average neural response to interspike intervals (ISIs) between paired stimuli (modified from [53]). ISIs were set according to the distance between the robot and its target and was set as positive if the target was to the right and negative if the target was to the left. Direction of movement was controlled by the ISI quadrants displayed by arrows. Speed of motion was controlled by the amplitude of the neural response.

Therefore, this relationship between interstimulus interval and response was used to manually create a mapping between the response and motion commands. Such an approach can be categorized as a supervised learning approach because the mapping between input and output is based on prior information. Neural learning was expected to be expressed through alterations in the behavior of the robot with time, such as changing its speed when approaching the target.

15.5.1.3 MEART: A Neurally Controlled Robotic Arm

A similar attempt that gained much attention was multi-electrode array art (MEART) [53], with which a robotic arm was manipulated via a disassociated neuronal network grown on a MEA. The goal of this experiment was to use the hybrid system to draw a portrait photograph, which was reduced to 60 pixels to simplify the task. Feedback in terms of the deviation between the actual photograph and the current state of the drawing was delivered to the neuronal network by altering the stimulation frequency of electrodes. The robotic arm movement was, in return, controlled by the network's response.

15.5.1.4 Other Attempts at Hybrid Systems

In applying this concept to a real world application, *in vitro* cultures were used to control a simulated aircraft's flight path, when variables such as altitude and roll were adjusted based on the response activity of the network, and the deviation of the actual flight path from the designated path was fed back into the network [54].

Supervised methods were also investigated by Warwick et al. [55] using a similar approach. Spikes were used as the basis of determining neuronal activity, and hence, real time spike detection was employed to implement a closed-loop learning system. Input–output mappings were created between pairs of electrodes with which one electrode delivered the stimuli, and the response was recorded from the other. Electrode pairs were selected for which the stimulation of one electrode biased the response of the other. Distances to obstacles were fed in through input electrodes by altering the stimulation frequency, and the robot was designed to turn if the corresponding output electrode registered activity within 100 ms. However, spontaneous responses from the output electrode also triggered unnecessary turns. In addition to obstacle avoidance, the right and left wheel speeds were also controlled by the spike rate of two output electrodes. The robot's collision avoidance abilities were seen to improve with time, implicating that plasticity or habitual performance was being instigated in the neuronal culture.

15.5.2 Unsupervised Learning

A common critique of supervised learning, as exemplified above, is that the link between neuronal activity and the set of output commands is constructed heuristically and with human intervention. Therefore, unsupervised methods that find structure within neuronal responses and create mappings automatically are sought.

Many of these methods employ self-organizing maps (SOMs) to find structure in the neural response and map it to actuator commands. SOMs, which are a type of artificial neural network (ANN), makes the visualization of response patterns easier because they map multidimensional inputs onto lower dimensions (generally two or three dimensions) while maintaining topological relationships between inputs. A SOM consists of a network of nodes onto which inputs are mapped, and the position of nodes change during training, depending on the features of these mapped inputs. Therefore, the final positioning of nodes and internode distances indicate relationships between inputs. SOMs further enable semisupervised learning, with which, if some of the classes of inputs are known, they can be used as seeds so that adjacent nodes are clustered onto these predetermined classes.

However, only SOMs with a constant number of nodes and a predefined node structure have been utilized with cultured neural networks. Growing SOMs [56], which incrementally increase the number of nodes during training, are a more dynamic alternative, which also resembles biological neural networks because it involves the concept of network growth. Therefore, it would be intriguing to investigate how a dynamic SOM would evolve in correspondence to the changes in real neuronal networks over time.

15.5.2.1 Inducing Plasticity through Training

The hypothesis that the same neuronal network could alter itself to perform different functions was tested by Bakkum et al. [57], using rat cortical cultures in an animat model. The animat was trained to move in one direction in a given training session, following which it was trained to move in another direction. The overall system is shown in Figure 15.10.

The mapping between neuronal response and motor output was created by considering the center of activity (CA), which is calculated using the location of electrodes relative to the center of the MEA and the firing rates recorded at each electrode. By considering the angle of the CA, the direction of motion for the animat was determined. The CA was also transformed by a matrix $\hat{T}$ to remove any bias from neurons having different spatial distributions.

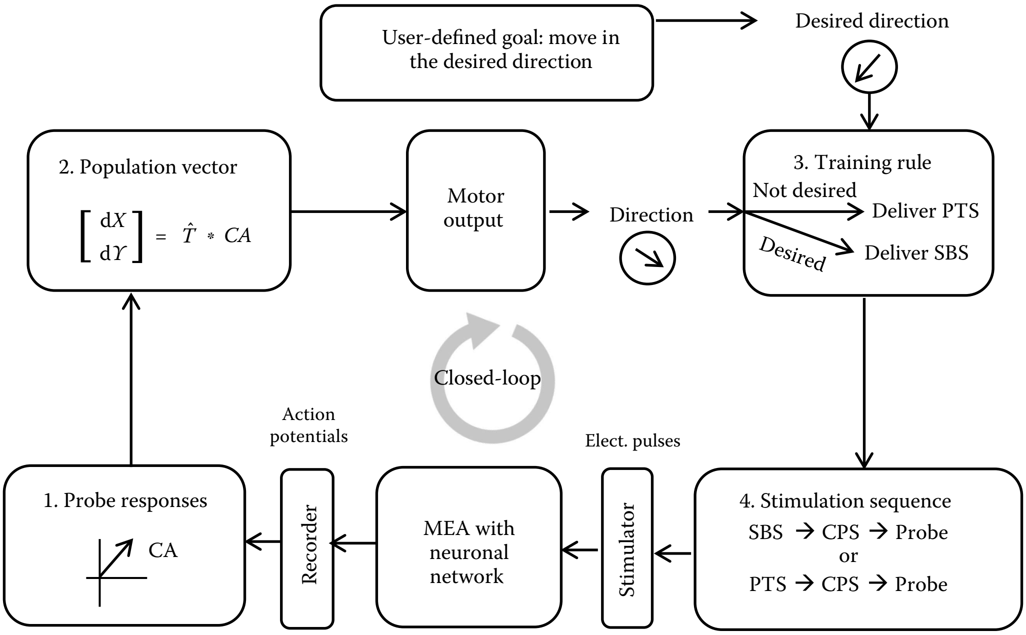

User-defined goal: move in the desired direction
Desired direction
2. Population vector
$\begin{bmatrix} dX \\ dY \end{bmatrix} = \hat{T} * CA$
Motor output
Direction
3. Training rule
Not desired
Deliver PTS
Desired
Deliver SBS
Closed-loop
Action potentials
Elect. pulses
1. Probe responses
CA
Recorder
MEA with neuronal network
Stimulator
4. Stimulation sequence
SBS → CPS → Probe
or
PTS → CPS → Probe

FIGURE 15.10
Closed-loop training process (modified from [57]). (1) After the delivery of the CPS stimuli, the generated response is used to build the center of activity (CA) vector. (2) CA is transformed into incremental movement [dX, dY]. (3) If the movement is in the desired direction, SBS is applied. Otherwise, PTS is repetitively applied. (4) CPS is again delivered after the PTS or SBS.

The hypothesis in this study was that different network states correspond to different modes of operation, and inducing plasticity means that the network was taken to a specific state and hence a specific operational mode. The training method was specifically designed to break existing network states and induce plasticity by creating new states. The stimulation sequence that was used to generate the desired motor output was termed a context–control probing sequence (CPS), and it included a sequence of pulses followed by a single "probe" pulse to generate a response reflecting the current functional state of the network. Because spontaneous bursting is a common dynamic in cultured networks, the series of pulses prior to the probe pulse ensured that bursting behavior was removed before the probe pulse was applied.

Patterned training stimuli (PTS), which consisted of pulses emitted from different electrodes were used to induce plasticity by altering the state of the network. A single pattern was applied multiple times to ensure that the network state was changed. Different patterns were assumed to induce different states. Hence by altering the PTS, the authors sought to find the optimum network state that drove the desired motor behavior. Shuffled background stimuli (SBS) were also used, in order to stimulate the network randomly without inducing a specific network state.

Therefore, if an application of the CPS successfully invoked the desired movement, the shuffled stimuli were applied afterward whereas, if movement was unsuccessful, different PTS was applied until the required movement was obtained. The probability of selecting a PTS that successfully invoked a desired response was given a higher probability of being selected in subsequent PTS selections while the opposite was applied to unsuccessful PTSs. This creates the feedback path of the system and makes the training method adapt to different target directions.

The experiments focused on a single probe stimulus in order to avoid response overlaps that would have occurred if multiple stimuli were used. Such response overlaps increase the complexity of the response and hence call for complex mappings between the response and motor output.

The successful adaptation of the network to move the animat in a given direction in most trials indicated that short-term plasticity was induced in the network. Closed-loop training was found to gradually lead the direction of movement of the animat toward the desired direction, and the learned direction was maintained after training when shuffled stimuli was applied, affirming that the desired response was indeed due to induced plasticity. The authors also found that plasticity was not induced in an open-loop condition, which emphasizes the need for feedback mechanisms in the formation of functional networks.

15.5.2.2 Use of Artificial Neural Networks for the Neural–Robotic Interface

The application of fully unsupervised learning to neurobots was investigated by Pizzi et al. [58], and an ANN called ITSOM (inductive tracing self-organizing map) was used as the interface between responses from human fetal neural stem cells and actuator outputs. They attempted to prove that neuronal networks can learn to respond selectively to different stimulation patterns. The authors used training sequences of stimulations to trigger neural activity, which was, in turn, fed into the ANN. The ANN was used as a classification system to relate the time-varying neural response with the input sequence. Four input patterns were used, and 10 runs each of the input patterns were used to initially train the ANN. Stimulations utilized multiple electrodes and the input patterns were chosen to provide sufficient complexity and good distinction between patterns (Figure 15.11). Four hundred milliseconds of neural response immediately following stimulation was used

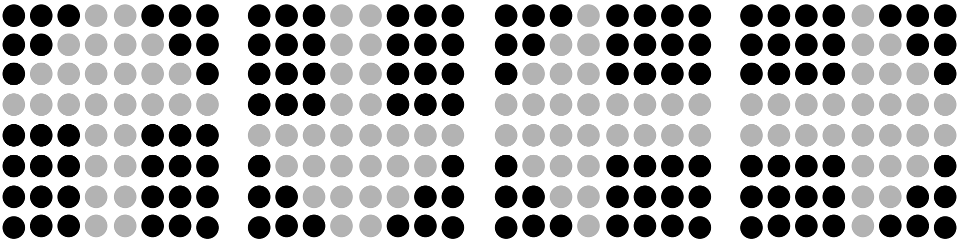


FIGURE 15.11
Four arrow-shaped patterns used for stimulating the culture (modified from [58]).

in training the ANN. Responses beyond this were not used because it was observed that neural activity returns to native state beyond this.

Training was followed by a test phase in which sequences from the four classes were fed into the culture, and the output from the culture–ANN hybrid system was assessed to measure its performance. Tests of the system, prior to training, yielded random results of which only 32% of the classifications were correct. Following training, classification accuracy increased to 60%. However, the authors also noted that most of the wrong classifications occurred in the last samples of the test set, which led them to question whether this was due to cell weakening after prolonged electrical stimulation in a suboptimal environment. An alternate hypothesis was that, the memory of neurons was erased after stimulation by a large number of random test sequences.

Unsupervised learning was also investigated by Kudoh et al. [59] when a SOM was used to map response features to actuator outputs that control the direction of movement of the robot. Their hybrid robot, termed Vitroid, used rat hippocampal cultures combined with a two-wheel robot. Sensor inputs were used to feed back information about the direction (left-L or right-R) of obstacles via stimulation by a pair of electrodes. A prior training phase was used to select two seed nodes in the SOM that created an optimal separation between the two directions (Figure 15.12). Seeding ensured that neural responses that are similar to one of these seed nodes are mapped to neighboring nodes. Nodes were, in turn, mapped to motor output, which was used to maneuver the robot forward without collisions.

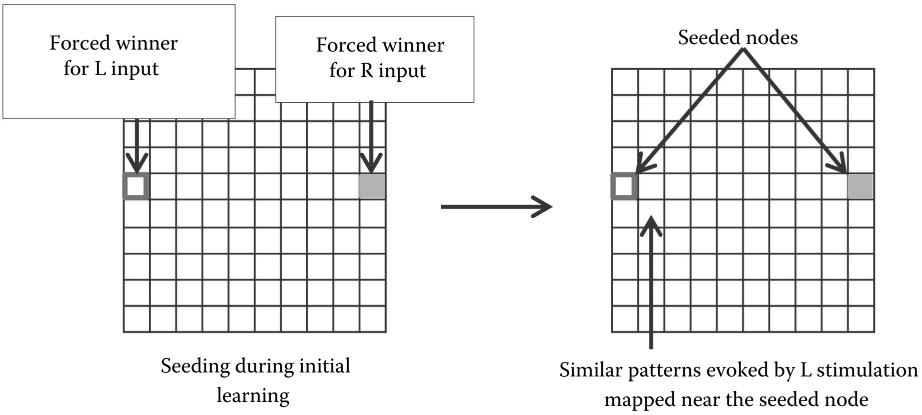

Forced winner for L input
Forced winner for R input
Seeding during initial learning
Seeded nodes
Similar patterns evoked by L stimulation mapped near the seeded node

FIGURE 15.12
Seeding the SOM during training. (From Kudoh, S. N., Y. Hukui, and H. Ito, Neurorobot Vitroid as a model of brain-body interaction, in *39th Annual Conference of the IEEE Industrial-Electronics-Society (IECON)*, 2003.)

15.6 Conclusion

In vitro MEAs are a flexible, high throughput technology of extracting spatiotemporal information of neuronal networks, whether it is spontaneous activity or activity patterns evoked by electrical stimuli or chemical perturbations. Understanding the complexity as well as the sources of network dynamics of intermediate-scale networks is of significant importance in building the bridge between microscopic and macroscopic operations in the brain. In particular, it paves the way for examining how neural plasticity scales from a single cell to a population of interconnected neurons.

Embodying neurons by coupling them with artificial actuators provides an intuitive mechanism of inducing plasticity and learning in the otherwise isolated neuronal networks. Embodiment attempts to study the ability of neurons to adjust local parameters, such as synaptic connections, and strengths to attain network level synergy for performing higher-level tasks. If dissociated cultures were embodied during early developmental stages, the growth of the dendritic arbor of neurons, which is believed to correspond to memory and learning, could be effectively monitored.

Such integrations of artificial and biological systems may yield invaluable benefits for both fields. This chapter focused on one direction of this relationship in which artificial systems aid in enhancing the understanding of biological systems. However, increased knowledge of the dynamic and complex functionality of neural systems would, in turn, lead to bio-inspired artificial systems as well as better compatibility between the two domains.

References

1. Ohkura, M. et al., Genetically encoded green fluorescent Ca^{2+} indicators with improved detectability for neuronal Ca^{2+} signals. *PLoS One,* 2012. **7**(12): p. e51286.
2. Akerboom, J. et al., Optimization of a GCaMP calcium indicator for neural activity imaging. *J Neurosci,* 2012. **32**(40): pp. 13819–40.
3. Hochbaum, D. R. et al., All-optical electrophysiology in mammalian neurons using engineered microbial rhodopsins. *Nature Methods*, 2014. 11: pp. 825–33.
4. Xie, Y. F., M. F. Jackson, and J. F. Macdonald, Optogenetics and synaptic plasticity. *Acta Pharmacol Sin,* 2013. **34**(11): pp. 1381–85.
5. Beyeler, A., C. A. Eckhardt, and K. M. Tye, Deciphering memory function with optogenetics, in *Molecular Basis of Memory,* 2014, **122**: pp. 341–90.
6. Leondopulos, S. S. et al., Chronic stimulation of cultured neuronal networks boosts low-frequency oscillatory activity at theta and gamma with spikes phase-locked to gamma frequencies. *J Neural Eng,* 2012. **9**(2): p. 026015.
7. Johnstone, A. F. et al., Microelectrode arrays: A physiologically based neurotoxicity testing platform for the 21st century. *Neurotoxicology,* 2010. **31**(4): pp. 331–50.
8. Illes, S. et al., Development and pharmacological modulation of embryonic stem cell-derived neuronal network activity. *Exp Neurol,* 2007. **207**(1): pp. 171–6.
9. Chiappalone, M. et al., Networks of neurons coupled to microelectrode arrays: A neuronal sensory system for pharmacological applications. *Biosens Bioelectron,* 2003. **18**(5–6): pp. 627–34.
10. Stett, A. et al., Biological application of microelectrode arrays in drug discovery and basic research. *Anal Bioanal Chem,* 2003. **377**(3): pp. 486–95.
11. Multichannel Systems, MEA-System: Extracellular recording with microelectrode arrays for all applications. Available at http://www.multichannelsystems.com/.

12. Spira, M. E. and A. Hai, Multi-electrode array technologies for neuroscience and cardiology. *Nat Nanotechnol,* 2013. **8**: pp. 83–94.
13. Gandolfo, M. et al., Tracking burst patterns in hippocampal cultures with high-density CMOS-MEAs. *J Neural Eng,* 2010. **7**(5): p. 056001.
14. Berdondini, L. et al., A microelectrode array (MEA) integrated with clustering structures for investigating in vitro neurodynamics in confined interconnected sub-populations of neurons. *Sens Actuators B-Chem,* 2006. **114**(1): pp. 530–41.
15. Egert, U. et al., MEA-Tools: an open source toolbox for the analysis of multi-electrode data with MATLAB. *J Neurosci Methods,* 2002. **117**(1): pp. 33–42.
16. Meier, R. et al., FIND—A unified framework for neural data analysis. *Neural Networks,* 2008. **21**(8): pp. 1085–93.
17. Wagenaar, D., T. B. DeMarse, and S. M. Potter, MeaBench: A toolset for multi-electrode data acquisition and on-line analysis, in *2005 2nd Internatinoal IEEE/EMBS Conference on Neural Engineering,* 2005, pp. 518–21.
18. Bologna, L. L. et al., Investigating neuronal activity by SPYCODE multi-channel data analyzer. *Neural Netw,* 2010. **23**(6): pp. 685–97.
19. Pasquale, V. et al., Self-organization and neuronal avalanches in networks of dissociated cortical neurons. *Neuroscience,* 2008. **153**(4): pp. 1354–69.
20. Eytan, D. and S. Marom, Dynamics and effective topology underlying synchronization in networks of cortical neurons. *J Neurosci,* 2006. **26**(33): pp. 8465–76.
21. Raichman, N. and E. Ben-Jacob, Identifying repeating motifs in the activation of synchronized bursts in cultured neuronal networks. *J Neurosci Methods,* 2008. **170**(1): pp. 96–110.
22. Chiappalone, M. et al., Burst detection algorithms for the analysis of spatio-temporal patterns in cortical networks of neurons. *Neurocomputing,* 2005. **65–6**: pp. 653–62.
23. Tokdar, S. et al., Detection of bursts in extracellular spike trains using hidden semi-Markov point process models. *J Comput Neurosci,* 2010. **29**(1–2): pp. 203–12.
24. Wagenaar, D. A., J. Pine, and S. M. Potter, An extremely rich repertoire of bursting patterns during the development of cortical cultures. *BMC Neurosci,* 2006. **7**: pp. 11.
25. Canepari, M. et al., Experimental analysis of neuronal dynamics in cultured cortical networks and transitions between different patterns of activity. *Biol Cybern,* 1997. **77**(2): pp. 153–62.
26. Kamioka, H. et al., Spontaneous periodic synchronized bursting during formation of mature patterns of connections in cortical cultures. *Neurosci Lett,* 1996. **206**(2–3): pp. 109–12.
27. Marom, S. and G. Shahaf, Development, learning and memory in large random networks of cortical neurons: lessons beyond anatomy. *Q Rev Biophys,* 2002. **35**(1): pp. 63–87.
28. Segev, R. et al., Hidden neuronal correlations in cultured networks. *Phys Rev Lett,* 2004. **92**(11): pp. 118102.
29. Baruchi, I. et al., The emergence and properties of mutual synchronization in in vitro coupled cortical networks. *Eur J Neurosci,* 2008. **28**(9): pp. 1825–35.
30. Maccione, A. et al., A novel algorithm for precise identification of spikes in extracellularly recorded neuronal signals. *J Neurosci Methods,* 2009. **177**(1): pp. 241–9.
31. Hulata, E., R. Segev, and E. Ben-Jacob, A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. *J Neurosci Methods,* 2002. **117**(1): pp. 1–12.
32. Hermle, T., C. Schwarz, and M. Bogdan, Employing ICA and SOM for spike sorting of multi-electrode recordings from CNS. *J Physiol Paris,* 2004. **98**(4–6): pp. 349–56.
33. Franke, F. et al., High-density microelectrode array recordings and real-time spike sorting for closed-loop experiments: an emerging technology to study neural plasticity. *Front Neural Circuits,* 2012. **6**: p. 105.
34. Selinger, J. V. et al., Methods for characterizing interspike intervals and identifying bursts in neuronal activity. *J Neurosci Methods,* 2007. **162**(1–2): pp. 64–71.
35. Pasquale, V., S. Martinoia, and M. Chiappalone, A self-adapting approach for the detection of bursts and network bursts in neuronal cultures. *J Comput Neurosci,* 2010. **29**(1–2): pp. 213–29.
36. Esposti, F. and M. G. Signorini, Synchronization of neurons in micro-electrode array cultures. *Eur Phys J Special Topics,* 2008. **165**(1): pp. 129–35.

37. Bakkum, D. J. et al., Parameters for burst detection. *Front Comput Neurosci,* 2013. **7**: p. 193.
38. Lefew, W. R. et al., Evaluation of microelectrode array data using Bayesian modeling as an approach to screening and prioritization for neurotoxicity testing. *Neurotoxicology,* 2013. **36**: pp. 34–41.
39. Biffi, E. et al., A new cross-correlation algorithm for the analysis of "in vitro" neuronal network activity aimed at pharmacological studies. *J Neurosci Methods,* 2011. **199**(2): pp. 321–7.
40. Chiappalone, M. et al., Dissociated cortical networks show spontaneously correlated activity patterns during in vitro development. *Brain Res,* 2006. **1093**(1): pp. 41–53.
41. Rolston, J. D., D. A. Wagenaar, and S. M. Potter, Precisely timed spatiotemporal patterns of neural activity in dissociated cortical cultures. *Neuroscience,* 2007. **148**(1): pp. 294–303.
42. Pimashkin, A. et al., Spiking signatures of spontaneous activity bursts in hippocampal cultures. *Front Comput Neurosci,* 2011. **5**: p. 46.
43. Raghavan, M. et al., A study of epileptogenic network structures in rat hippocampal cultures using first spike latencies during synchronization events. *Phys Biol,* 2012. **9**(5): p. 056002.
44. Eckmann, J. P. et al., Leader neurons in population bursts of 2D living neural networks. *New J Phys,* 2008. **10**(1): p. 015011.
45. Ham, M. I. et al., Spontaneous coordinated activity in cultured networks: Analysis of multiple ignition sites, primary circuits, and burst phase delay distributions. *J Comput Neurosci,* 2008. **24**(3): pp. 346–57.
46. Lopes-Dos-Santos, V., S. Ribeiro, and A. B. Tort, Detecting cell assemblies in large neuronal populations. *J Neurosci Methods,* 2013. **220**(2): pp. 149–66.
47. Gullo, F. et al., Short-latency cross- and autocorrelation identify clusters of interacting cortical neurons recorded from multi-electrode array. *J Neurosci Methods,* 2009. **181**(2): pp. 186–98.
48. Fuchs, E. et al., The formation of synchronization cliques during the development of modular neural networks. *Phys Biol,* 2009. **6**(3): p. 036018.
49. Gullo, F. et al., Classifying heterogeneity of spontaneous up-states: A method for revealing variations in firing probability, engaged neurons and Fano factor. *J Neurosci Methods,* 2012. **203**(2): pp. 407–17.
50. Gullo, F. et al., Orchestration of "presto" and "largo" synchrony in up-down activity of cortical networks. *Front Neural Circuits,* 2010. **4**: pp. 11.
51. DeMarse, T. B. et al., The neurally controlled Animat: Biological brains acting with simulated bodies. *Auton Robot,* 2001. **11**: pp. 305–10.
52. Potter, S. M. et al., Long-term bidirectional neuron interfaces for robotic control and in vitro learning studies, in *IEEE EMBS,* 2003.
53. Bakkum, D. et al., Removing some 'A' from AI: Embodied cultured networks, in *Embodied Artificial Intelligence* Lecture Notes in Computer Science, 2004. 3139: pp. 130–45.
54. DeMarse, T. B. and K. P. Dockendorf, Adaptive flight control with living neuronal networks on microelectrode arrays, in *IEEE International Joint Conference on Neural Networks (IJCNN 2005),* Montreal, Canada, 2005.
55. Warwick, K. et al., Controlling a mobile robot with a biological brain. *Def Sci J,* 2010. **60**(1): pp. 5–14.
56. Alahakoon, D. and S. K. Halgamuge, Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Trans Neural Netw,* 2000. **11**(3): pp. 601–14.
57. Bakkum, D. J., Z. C. Chao, and S. M. Potter, Spatio-temporal electrical stimuli shape behavior of an embodied cortical network in a goal-directed learning task. *J Neural Eng,* 2008. **5**(3): pp. 310–23.
58. Pizzi, R. M. et al., A cultured human neural network operates a robotic actuator. *Biosystems,* 2009. **95**(2): pp. 137–44.
59. Kudoh, S. N., Y. Hukui, and H. Ito, Neurorobot Vitroid as a model of brain-body interaction, in *39th Annual Conference of the IEEE Industrial-Electronics-Society (IECON),* 2003.

Index

Page numbers followed by f and t indicate figures and tables, respectively.

A

Absolute encoders, 209–210, 210f
Accelerometers, 177, 178–181
  Bruel & Kjar charge accelerometer type 4501, 178, 178f
  charge, 178–179, 178f, 179f
  integrated electronic piezoelectric (IEPE) accelerometers, 177, 179, 179f, 180f
  MEMS, 188
  pair force transducer *vs.* accelerometer, calibration, 197–198, 197f
  piezoelectric, 177, 178–180, 178f, 179f, 180f
  piezoresistive and capacitive, 180–181, 181f
  T-shaped block technique, 180, 180f
Acoustic sensors, bio-MEMS, 245
Actions, *Q*-learning controller, 555–557, 555f
Active car suspension, model-referenced, 347–349, 347f, 349t, 350f
Active sensors, 177
ACTRESS, multi-robot system, 296
ACTS color blob-tracking software, 518, 546, 550, 557
ACTSTM software, 523
Actuator(s)
  bio-MEMS, 222–240
    artificial muscles, 222–226; *see also* Artificial muscles
    biomolecular systems, 240
    capillary valves in microfluidic devices, 233–235, 234f
    ciliary, 226–229, 227f, 228f
    DDS, 235–240; *see also* Drug delivery system (DDS)
    nanotweezers, 229–233; *see also* Nanotweezers
  for control of rigid bodies in 3-D space
    modeling, 18, 26–30
    two-mass spring system and, MBDS, 33, 33f
  dynamic force, 466
  dynamics, 90, 91
  in flight simulators, 114–115
  force by, 464–465
  linear, 461
  linear dynamics, 91
  management, in unmanned aircraft system, 415–416
  piezoelectric, 237
  regenerative, 454, 462
Adaptive burst detection methods, 576
Adaptive control, ICSI system and, 122, 122f
Adaptive nonlinear model predictive control (A-NMPC), 518, 522–525, 522f, 524f–526f
Add-element construction functions, 329, 329f, 330
Aeroelastic vibration
  for piezoelectric energy harvesting, 473–477, 474f
Aircraft wing flutter analysis, 474
Airships, 466–467, 467f
  behavior of, 468
  buoyancy force, 467
  external forces on, 467–468, 468f
  power system, 468–470, 469f–470f
  solar energy as power source for, 468–470, 469f–470f
AISIMAM model, 306
Aliasing, 150–154
  charge amplifier type 2635, 153, 153f
  defined, 152
  elimination, 154
  examples, 150–154, 151f, 152–153, 153f, 154f
  prevention, 152–153
  sampling frequency and, 151–153, 151f
Amorphous silicon cells (a-Si), 470
Amplitude quantities, 211, 213–214
Analog sensors, 177
Analog-to-digital conversion (ADC)
  quantization errors, 155
  sampling frequency in, 147–148
Anemometers, 203–204, 204f
Angle of twist, of element, 70
Angular momentum flow meters, 204–205, 205f
"Animat" model, 579–580, 579f
Antibodies
  antibody–antibody paratope and idiotope matching, 307, 308f
  antibody–antibody stimulation and suppression, 306–307, 307f

- antibody–antigen stimulation and suppression, 306–307, 307f
- defined, 297
- elimination of antigen by, 298–299, 298f, 317, 317f, 318f
- failures, results of, 312–318
  - cooperating, 314f, 315f, 316
  - facilitating, 312, 312f, 313f, 315
  - instigating, 312, 313f, 314f, 315–316
- robot and, 306, 306f
- selecting appropriate, 311–312
- structure, 297–298, 298f

Antigens
- antibody–antigen stimulation and suppression, 306–307, 307f
- elimination, by antibodies, 298–299, 298f, 317, 317f, 318f
- epitope, 305, 306–307, 307f
- FAB region, 298

Antivibration design, UAV system, 417, 418f, 419f

Apparent mass, defined, 197

Arbitrator, of hybrid control system, 558

Arc, defined, 70

Argand plane, harmonic signal in, 130–132
- amplitude, 130, 130f
- circular vector rotating in, 131, 131f
- perpendicular orthogonal axes, 131

Arithmetic construction functions, 330

Array-wide bursting, MEAs, 574

Array-wide spike rate, 574

Artificial immune system (AIS)
- multi-robot cooperation and, 305–308
  - binding affinity, 305
  - modified idiotypic network model, 306–308, 307f, 308f
  - robot and antibody, 306, 306f
- network theory model, 300–301

Artificial intelligence (AI)
- Markov decision process in, 530–534

Artificial muscles, EAP for, 222–226
- dielectric, 225–226, 226f
- ionic, 223–225, 223f, 224f
- overview, 222–223

Artificial neural network (ANN), 581
- for neural–robotic interface, 583–584, 584f

Artificial vision, 501

Assembly, of equations of motion, 21–26, 22f

Asus Xtion PRO LIVE, 414

Asymmetrical load, freely supported beam with, 48–50, 48f

Atomic force microscopes (AFM), molecular manipulation by, 229, 229f

Attitude and heading reference system (AHRS), 409

Attributes
- Markov decision process (MDP), 531

Auto-correlation function, 162, 163f

Automated mechatronic design tool, 324–356
- controller design using BGs, 335–337
  - admittances, 336
  - collocated control, 335–336, 335f, 336f
  - impedances, 335, 336
  - representation of different control schemes, 337f
  - zero-junction and one-junction structures, 337
- design theory, 324, 325
- evolutionary mechatronic tool, 325–334
  - BGs, *see* Bond graphs (BGs)
  - GP, *see* Genetic programming (GP)
- general description, 324f
- niching optimization scheme, 345–356
  - GA, 345–346
  - GP, 346
  - hydraulic engine mount design, case study, 349, 351–356; *see also* Hydraulic engine mount design
  - model-referenced active car suspension, case study, 347–349, 347f, 349t, 350f
- overview, 324
- two-loop design model, 337–345
  - flowchart of natural evolution, 338f
  - hybrid GA with GP, 339–341, 340f
  - iron butcher controller design, case study, 341–345, 341f, 342f, 343f, 344f, 345f

Auto-spectral density (ASD), 162–163

Average, moving, 174–175, 175f

Average firing rate (AFR), 576

Averaging, ensemble, 167

Avionics, 405, 407

B

Backlash, 108–109, 108f

Band-pass filters, 168

Band-reject filters, 168

Barlett's method, 167

Base excitation
- mass-spring-damper system with, 455f

Beams
- bending strain, 51–52, 52f
- bending stress, 52–53
- deflection of, 55–69
  - bending moments, 57, 58f

        - discontinuous bending moment equations, 64, 64f
        - equilibrium and determinacy, 57, 57f, 58f
        - flexure equation, 56–57, 56f, 58, 59
        - overview, 55–56
        - singularity function (Macaulay's) method, 65–69, 65f, 67f, 68f, 69f
        - statically indeterminate beams, 62–64, 63f
        - transverse loaded slender, 59–62, 60f, 61f
    - freely supported
        - asymmetrical load, 48–50, 48f
        - symmetrical load, 47–48, 47f
        - uniform load, 50–51, 50f
- Bell, Alexander Graham, 211
- Bellman equation, 532
- Bending, theory, 45–55
    - defined, 46
    - moment and shearing force
        - beam deflections, 58, 58f
        - diagrams, 47–51
        - sign convention of, 46, 47f
    - overview, 45–46
    - planar bending model, 46, 46f
    - stresses, 51–55
        - assumptions, 51
        - beam bending, 52–53
        - beam bending strain, 51–52, 52f
        - section properties, 53–55, 53f, 54f
- Bending moments
    - beam deflections, 58, 58f
    - diagrams, 47–51; *see also* Diagrams, bending moment and shear force
    - discontinuity, 64, 64f
    - sign convention of, 46, 47f
- Bernoulli equation, 480
- BGs, *see* Bond graphs (BGs)
- Bicycle system, power produced by, 470–471, 471f
- Bimetallic DDS, 238, 239, 239f
- Bimetallic thermometers, 200, 200f
- Binding affinity
    - optimizing, function using GA, 310–311, 311t
    - between paratope and epitope, 300–301
    - in resolving conflicts, 301, 303, 305
- Bio-inspired dynamic systems, 471–494, 472f–473f
    - flapping wing flying robots, 481–488, 481f–486f, 488f
    - flight control system, 488–491, 489f, 491f
    - piezoelecteric energy harvesting from aeroelastic vibrations, 473–477, 474f
    - self-propelled vehicle, 478–481, 479f, 480f
    - uncertainty quantification, 491–494
    - VAWTs inspired by fish schooling, 477–478
- Bio-inspired flapping wing flying systems, *see* Flapping wing flying systems, bio-inspired
- Bio-inspired flight control system, *see* Flight control system, bio-inspired
- Bio-inspired piezoelectric energy harvester, 472–473, 472f–473f
- Bio-inspired self-propelled vehicle, 478–481, 479f–480f
    - pulsed-jet propulsion, 479, 480–481, 480f
    - steady-jet propulsion, 479–480, 480f
- Biological immune system, 297–299, 298f
    - Jerne's idiotypic network theory, 299, 299f
- Biological systems, 454, 471
- Biomedical application, piezoelectric control system for, 116–122
    - ICSI installation, design of, 116–122, 116f
        - adaptive control, 122, 122f
        - linear reciprocating motion, 117
        - LVDT, 117–118, 118f
        - oocytes, structure of, 116–117, 116f
        - PID control, 119–121, 120f
        - Simulink®, 118, 119f
        - system identification, 119
- Biomedical MEMS (bio-MEMS), 221–245
    - actuators, 222–240
        - artificial muscles, 222–226; *see also* Artificial muscles
        - biomolecular systems, 240
        - capillary valves in microfluidic devices, 233–235, 234f
        - ciliary, 226–229, 227f, 228f
        - DDS, 235–240; *see also* Drug delivery system (DDS)
        - nanotweezers, 229–233; *see also* Nanotweezers
    - overview, 222
    - sensors, 240–245
        - acoustic, 245
        - blood cell counter, 244, 244f
        - for CRP detection, 241, 242, 242f
        - force sensor for protein delivery, 243, 243f
        - glucose detection, 242, 242f
        - tissue softness characterization, 243, 244, 244f
        - triglyceride biosensor, 241, 241f
- Biomolecular systems, 240
- Birds
    - flight control parameters, 488–489, 489f
    - formations in flight of, 490–491, 492f

- optimum value of wingtip spacing (WTS_{opt}), 491
- visual control, 488

Blood cell counter, 244, 244f

Bode diagram
- of HO model, 267f
- in iron butcher controller design, 344–345, 344f
- model-referenced active car suspension system, 350f

Bond graphs (BGs)
- causality stroke
  - bond between elements with, 362, 362f
  - defined, 336, 362
- controller design using, 335–337
  - admittances, 336
  - collocated control, 335–336, 335f, 336f
  - impedances, 335, 336
  - representation of different control schemes, 337f
  - zero-junction and one-junction structures, 337
- embryo model, 328, 329f, 334, 334f, 351, 352f
- evolutionary design framework with, 372–374, 372f, 373f, 373t
- evolutionary mechatronic tool, 327–328
- GP and, integration, 328–334
  - add-element construction functions, 329, 329f, 330
  - arithmetic construction functions, 330
  - construction functions and terminals, 329–330
  - eigenvalue assignment, case study, 333–334, 334f, 334t
  - embryo model, 328, 329f
  - first generation, 330–331, 331f
  - insert-junction construction functions, 330, 330f
  - new generations, creation of, 332–333, 332f, 333f
- model, hydraulic engine mount design and
  - double-notch, 351, 352f
  - elite solutions, 352, 353f–354f, 355
  - embryo model, 351, 352f
  - typical single pumper, 349, 350f
- modeling, 361–368
  - causality assignment, 363–364, 364f, 364t
  - electrical and mechanical system, 362, 363, 363f
  - elements, 361, 362f
  - state-space model, 365–368, 365t, 366f
  - variables, 362, 362t

Bragg cell, 181

Brake mode, regenerative system, 464

Bruel & Kjar charge accelerometer type 4501, 178, 178f

BumbleBeeR2 stereo camera, 506, 506f

Buoyancy force, in airship, 467

Bursts, in MEA recordings, 573, 574f
- detection of, 576, 576f

Butterworth filters, 168–173
- amplification factor, 172–173, 173f
- generalizing, 169
- low-pass filter gain amplitude, 169f
- op-amp circuit for, 171, 172, 172f, 174f
- polynomial coefficients, 170
- on time and frequency domains, 171f

C

Cadmium sulphide (CdS), 470

Calcium imaging, 570

Calibration
- factor, defined, 177
- pair force transducer *vs.* accelerometer, 197–198, 197f

Camera
- projection model, vision-based mobile robot, 548–549, 548f
- RGB-D, 413–415

Camera, visual servo systems, 510–513, 511f–512f
- configurations
  - camera-in-hand, 504–505, 504f
  - eye-to-hand, 504–505, 504f
  - mobile robotic visual servo systems, 504–505, 504f
- extrinsic parameters, 513
- intrinsic parameters, 514
- modeling
  - center of projection, 510
  - image plane, 510–512, 511f, 512f
  - pinhole, 510–511, 511f
  - pixels, 511, 512f
  - principal point, 510
- parameters, 513–514

Camera frame, of mobile robot, 514, 516f, 547–548, 547f, 548f

Camera-in-hand configuration, 504–505, 504f

Cantilever beam
- point load at free end, 59–61, 60f
- uniformly distributed load with unit length, 61–62, 61f

Capacitance force, 459
Capacitive accelerometer, 180–181, 181f
Capillary valves, in microfluidic devices, 233–235, 234f
Cargo transportation system, UAV, *see* Unmanned aerial vehicle (UAV)
Carrier frequency, defined, 181, 182
Case studies
  coaxial rotorcraft system, 424–436; *see also* Coaxial rotorcraft system
  eigenvalue assignment, 333–334, 334f, 334t
  hydraulic engine mount design, 349, 351–356; *see also* Hydraulic engine mount design
  iron butcher controller design, 341–345, 341f, 342f, 343f, 344f, 345f
  model-referenced active car suspension, 347–349, 347f, 349t, 350f
  UAV cargo transportation system, 436–447
Causality assignment, in BG modeling, 363–364, 364f, 364t
Causality stroke
  bond between elements with, 362, 362f
  defined, 336, 362
Cayley-Hamilton theorem, 289
CCD camera, 518, 525, 545–546
Center of activity (CA), MEA and, 582
Center of gravity (COG)
  acceleration vector of, 20
  defined, 20
Central nervous system (CNS), 568
Charge accelerometers, 178–179, 178f, 179f
Charge amplifier
  defined, 179
  type 2635, 153, 153f
Charge-coupled device (CCD), 517
Chebishev filters, 168
Ciliary actuators, 226–229, 227f, 228f
Clonal expansion, defined, 299
Closed-loop training process, 582
  MEA and, 582–583, 582f
Clustering, of neurons, 578
Coaxial rotorcraft system, case study, 424–436
  experimental results, 434, 435f, 436
  hardware system, 424–431
    communication unit, 429
    computers, 427, 428
    control hub, 429
    integration, 429, 431, 431f
    navigation sensors, 427, 428f, 428t
    overview, 424–427, 425f, 425t, 426f
    servo controller, 428–429
  software system, 432–434
    ground control software system, 433, 434, 435f
    onboard real-time software system, 432–433, 432f, 433f, 434f
Coefficients of thermal expansion (CTE), 226–228
Cogging force, 107
Collocated control methodologies, 335–336, 335f, 336f
Communication units, in unmanned aircraft system, 416–417, 416f, 417t
  coaxial rotorcraft system, 429
Compatibility equations, in LGs, 369, 371
Complementary shear stress, 41–42, 42f
Complex harmonic signals, 129
Compression tests, property of materials, 80, 80f, 81f
Compressive strain, 43
Compressive stress, 39, 40–41, 40f
Computer(s)
  control, implementation, 103–106, 104f, 105f
  in unmanned aircraft system, 415
    coaxial rotorcraft system, 427, 428
    computer-aided virtual design, 417, 418f, 419f
    UAV cargo transportation system, 441
Condition monitoring, 454
Constant percentage bandwidth filters, 216
Constitutive equations, in LGs, 369, 370f, 371
Construction functions and terminals
  BGs and GP, integration, 329–330
    add-element functions, 329, 329f, 330
    arithmetic functions, 330
    insert-junction functions, 330, 330f
Context–control probing sequence (CPS), 583
Continuity equations, in LGs, 369, 371
Control hub, in coaxial rotorcraft system, 429
Control law, vision-based mobile robot motion control, 549–550
Controller, LQG
  defined, 269
  with time delay, 271–274, 274f
  without time delay, 269–271
Controller design
  iron butcher, case study, 341–345, 341f, 342f, 343f, 344f, 345f
  using BGs, 335–337
    admittances, 336
    collocated control, 335–336, 335f, 336f
    impedances, 335, 336

- representation of different control schemes, 337f
- zero-junction and one-junction structures, 337

Control(s)
- DCSs, implementation
  - advantage of, 103
  - computer-based, 103–106, 104f, 105f
- HAM
  - manual control, 254–255
  - optimal control theory, *see* Optimal control theory
- of mechatronic systems, 88–122; *see also* Mechatronic systems, control
- of rigid bodies in 3-D space, *see* Rigid bodies in 3-D space

Control scheme
- hybrid controller for robust visual servoing, 546, 546f
- vision-based mobile robot motion control, 546, 546f

Control theory, 499

Conveying system, IIB
- design improvement of, 398, 400f
- modeling and simulation, 386–394
  - electromechanical, 387–389, 388f
  - gearbox, 387–389, 392f
  - lever, 387–389
  - push–pull movement, 393f
  - SimMechanics model, 391, 393, 394f
  - sliding mechanism, 387f
  - state-space model, 387–391
  - VDP drive, 386, 387, 391
  - VFD, 391, 391f
  - wheel/axle, 387–389

Convolution
- integers, defined, 175
- in signal processing, 160–161, 160f, 161f

Cooperating antibody, partial and full failures, 314f, 315f, 316

Cooperation strategy, multi-robot transportation system, 536–541
- genetic algorithms, 538–540, 538f
- reinforcement learning, 536–538, 537f
- reinforcement learning and genetic algorithms, integration scheme, 540–541, 541f

Coriolis effect, 188

Coriolis principle, 206

Cost function, defined, 265

Coupled design, 5–7, 6f

C-reactive protein (CRP) detection, bio-MEMS sensors for, 241, 242, 242f

Crossover, genetic operation, 309, 309f, 376

Crossover frequency, HO model and, 258–259

Crossover model, 257, 258–259

Cross-spectral density (CSD), 164–165

Cruisers, 467

Crystalline silicon cells, 470

C-scan, principle of operation, 207, 208f

CSD (cross-spectral density), 164–165

CTLs (cytotoxic T lymphocytes), defined, 297

Culture-ANN hybrid system, 584

Cultured neural networks (CNNs), 571f

Current-voltage relationship
- of PV cells, 469–470, 470f

Cytoplasm, oocytes, 116–117, 116f

Cytotoxic T lymphocytes (CTLs), defined, 297

D

Damping ratio
- dimensionless power *vs.*, 457–458, 458f
- harvested power *vs.*, 458

Damping systems, linear momentum force systems, 19–20

Dangerous area, hybrid visual servoing with
- large, 560–561, 561f–564f
- small, 558–559, 559f, 560f

DARPA Urban Challenge, 501–502, 502f

Data-Based HO model identification, 289–292, 290f, 291f, 292f

DDS, *see* Drug delivery system (DDS)

Dead zone model, 109, 110f

Decibel (dB), 211–215
- dB power *vs.* root-power ratio scales, 214, 215t
- linear *vs.* logarithmic frequency plots, 214, 214f
- power (intensity) quantities, 211, 212–213
- reference values, 214, 215t
- root-power (amplitude) quantities, 211, 213–214
- sound-level meter, 212f

Decision conflicts, multi-robot cooperation problem, 302–303

Deflection, of beams, 55–69
- bending moments, 57, 58f
- discontinuous bending moment equations, 64, 64f
- equilibrium and determinacy, 57, 57f, 58f
- flexure equation, 56–57, 56f, 58, 59
- overview, 55–56
- singularity function (Macaulay's) method, 65–69, 65f, 67f, 68f, 69f
- statically indeterminate beams, 62–64, 63f

- transverse loaded slender, 59–62
  - cantilever beam, point load at free end, 59–61, 60f
  - uniformly distributed load with unit length, 61–62, 61f

Deformation, 42

Denavit-Hartenberg (DH) convention, 508–509, 509t
- coordinate frames, 509, 509f

Derivative causality, in BG modeling, 363

Design evolution, 371–401
- application of methodology to industrial systems, 377–400
  - IIB, *see* Intelligent iron butcher (IIB)
- of engineering systems using GP, 371–372
- examples, 377, 378f, 379f
- fitness function, 375–377, 375f, 376f
- framework with BGs, 372–374, 372f, 373f, 373t
- GA in, 371
- methodology, 374
- overview, 8–9, 9f, 360–361, 361f, 371–372
- solution representation, 374–375, 374f

Designs, 3–9, 4f
- coupled, 5–7, 6f
- evolution, 8–9, 9f
- GA, 7
- index, 6
- MDQ, 7–8
- modeling and, 3–4, 4f
- problems, 5
- of unmanned aircraft systems; *see also* Unmanned aircraft systems
  - coaxial rotorcraft system, case study, 424–436
  - UAV cargo transportation system, case study, 436–447

Desirable area, on image, 559

Determinacy, beam deflections, 57, 57f, 58f

Deterministic signals, 128, 129

DH (Denavit-Hartenberg) convention, 508–509, 509t
- coordinate frames, 509, 509f

Diagrams, bending moment and shear force
- freely supported beam
  - asymmetrical load, 48–50, 48f
  - symmetrical load, 47–48, 47f
  - uniform load per unit run, 50–51, 50f

Dielectric EAP, 225–226, 226f

Differential GPS (DGPS), 410

Differentiation, of harmonic signals, 132–134, 133f

Digital sensors, 177

Digital signal processing (DSP), 160

Dimensionless power, 464
- *vs.* frequency ratio and damping ratio, 457–458, 458f

Dirac δ-function, 160, 161f

Direct strains, 42–43
- compressive, 43
- tensile, 42

Direct stresses, 39–41
- compressive, 39, 40–41, 40f
- tensile, 39–40, 40f

Dirichlet condition, for infinite random signals, 162

Discontinuous bending moment equations, beams with, 64, 64f

Discrete Fourier Transform (DFT), 148–149

Displacement transducers, 184–189
- lasers, 186–187, 186f
- LVDTs, 184–185, 185f
- MEMS sensors, 188–189, 188f
- proximity probes, 187–188, 187f

Displacement vector
- uncertainty quantification (UQ), 492

Distributed control systems (DCSs), implementation
- advantage of, 103
- computer-based, 103–106, 104f, 105f

DLR Rollin' Justin, 502, 503f

Dolby noise reduction system, 168

Doppler vibrometer, 241, 241f

Double-notch hydraulic engine mount, 351, 352f

Drive mode, regenerative system, 464

Drive rods, 194, 194f

Drug delivery system (DDS), 235–240
- bimetallic, 238, 239, 239f
- electromagnetic, 239, 240f
- electrostatic, 236, 237f
- ICPF, 239, 239f
- micropumps
  - electrostatic actuation mechanism in, 236
  - mechanical, 235
  - nonmechanical, 235
- piezoelectric, 237, 237f
- schematic representation, 235, 236f
- SMA, 238, 238f
- thermo-pneumatic, 238, 238f

DSPACE control development, 118

Ductile materials, stress–strain behavior of, 80, 81, 81f

Duhamel's integral, 160–161, 160f, 161f

Dynamic force
- regenerative system, 466

Dynamic friction, in mechatronic control system, 106

Dynamic systems
  bio-inspired, *see* Bio-inspired dynamic systems
  displacement vector, 492–493
  renewable energy for, 466–470, 467f, 468f–470f
  self-powered, *see* Self-powered dynamic system

E

Effort variable, in BG model, 362, 362t
E – greedy search policy, *Q* learning, 533–534
Eigenvalue assignment, case study, 333–334, 334f, 334t
Elasticity
  Hooke's law and elastic constants, 44–45
  modulus of, 45
  stress and strain, 38–45; *see also* Strains; Stresses
  yield and, 44
Elastic limit, defined, 44, 80
Electrical energy
  internal, 475
Electrical power, in coupled design, 5–6
Electric enthalpy density, 460
Electric motors
  concept of, 461
  torque constant, 461
  voltage constant, 461
Electroactive polymers (EAP), for artificial muscles, 222–226
  dielectric, 225–226, 226f
  ionic, 223–225, 223f, 224f
  overview, 222–223
Electroencephalography (EEG), 570
Electromagnetic DDS, 239, 240f
Electromagnetic energy
  conversion mechanism, 461, 461f, 464
    regenerative system, 463–464, 463f; *see also* Self-powered dynamic system
Electromagnetic interference (EMI) shielding design, 421
Electromagnetic proximity probes, 184, 184f
Electromagnetic pulse tachometer, 208, 209f
Electromagnetic regenerative actuators, 463–464
Electrostatic energy
  conversion mechanism, 459, 459f
Elliptic (equiripple) filters, 168
Embodied neuronal networks
  supervised learning, 579–581
    "animat" model, 579–580, 579f
    hybrid systems, 581
    hybrot, 580–581, 580f
    MEART, 581
  unsupervised learning, 581–584
    ANNs for neural–robotic interface, 583–584, 584f
    inducing plasticity through training, 582–583, 582f
Embryo models, BG, 328, 329f, 334, 334f, 351, 352f, 374–375, 374f
Encoders
  absolute, 209–210, 210f
  incremental, 208–209, 209f
  overview, 208
Energy conversion mechanisms, 459–461, 466
  electromagnetic, 461, 461f, 464
  electrostatic, 459, 459f
  piezoelectric, 459–461, 460f
Energy harvester/harvesting, 454
  design, 454
  low-frequency, 454
  mechanisms, 454
  piezoelectric, 455
  in water distribution systems, 454
Energy harvesting techniques
  concept of, 455–456, 455f
  conversion mechanisms, 459–461, 459f–461f
    electromagnetic energy, 461, 461f
    electrostatic energy, 459, 459f
    piezoelectric energy, 459–461, 460f
  kinetic energy for, 455, 456
  mass-spring-damper system, 455–456, 455f
  overview, 454
  for self-powered dynamic system, 455–461; *see also* Self-powered dynamic system
  sources, 455
Engineering constants, 189
Engine mount design, hydraulic, 349, 351–356
  BG model
    double-notch hydraulic engine mount, 351, 352f
    elite solutions, 352, 353f–354f, 355
    embryo model, 351, 352f
    typical single pumper, 349, 350f
  dynamic stiffness, 351, 351f, 352, 355, 355f, 356f
Ensemble averaging, 167
Epitope
  of antibody, 298, 299, 300–301
  antigen, 305, 306–307, 307f
  defined, 298
Equations of moment of momentum, 21
Equations of motion
  assembly of, 21–26, 22f

  for linear model, 19
  of mass-spring-damper system, 455–456
Equilibrium, beam deflections, 57, 57f, 58f
Ergodic signal, 162
Errors, quantization, 155, 155f
Estimators, 165–167, 166f, 167f
Evolution
  design, *see* Design evolution
  of mechatronics, 10
  mechatronic tool, 325–334
    BGs, *see* Bond graphs (BGs)
    GP, *see* Genetic programming (GP)
Evolutionary learning mechanism, 541–543, 542f–543f
Expert system, for mechatronic design, 8–9
Exponential ensemble averaging, 167
Exponential window, 159
Extensometer, measuring strain with, 189–190, 189f, 190f
Eye-to-hand camera configuration, 504–505, 504f

F

Facilitating antibody, partial and full failures, 312, 312f, 313f, 315
Facilitating robot, defined, 301
Failures, antibodies, 312–318
  full, 312, 312f, 313f, 314f, 315–316, 315f
  partial, 312, 312f, 313f, 314f, 315–316, 315f
Far infrared (FIR) region, 201
Fast Fourier transform (FFT), 149–150, 149f
Fault tolerance, multi-robot systems and, 302
Feeders, 467
Feedforward control, 101, 101f
Filters
  band-pass, 168
  band-reject, 168
  butterworth, *see* Butterworth filters
  Chebishev, 168
  elliptic (equiripple), 168
  FIR, 168
  IIR, 168
  linear continuous time, 168
  low-pass, 168
  smoothing, *see* Smoothing filters
FIND, 573
Finite impulse response (FIR), filters, 168
Firing rate, 573
Fish processing industry, IIB for, *see* Intelligent iron butcher (IIB)
Fish schooling, 472, 472f
  vertical axis wind turbine farm inspired by, 477–478
Fitness function
  defined, 333–334
  design evolution, 375–377, 375f, 376f
  IIB, 383–384, 383f
Flapping wing flying systems, bio-inspired, 481–488, 481f–486f, 488f
  actuation moment, 486–487, 486f
  angle of attack, 482, 482f, 485f
  drag force *vs.* time, 482, 484, 484f
  flapping angle, 482, 482f, 485f
  force applied, 481–482, 481f
  lift and drag forces *vs.* time, 484–485, 484f
  lift force for hovering *vs.* frequency of flapping, 487–488, 488f
  lift force *vs.* angle of attack and flapping angle, 482–484, 483f
  lift force *vs.* time, 482, 483f
  parameters of, 481f
Flexure equation
  beam deflections, 56–57, 56f, 58, 59
  defined, 53
Flight control system, bio-inspired, 488–491, 489f, 491f
  actuators in, 489
  block diagram, 489–490, 490f
  formation flight of birds and, 490–491, 492f
  objectives of, 489
Flight of birds, formations in, 490–491, 492f
Flight simulators, 114–116
  actuators, framework of, 114–115
  disturbance signals in, 115, 115f
  motion commands, 115
  power moderation, role, 114
Flow rate meters, 202
Flow sensors, 201–206
  anemometers, 203–204, 204f
  angular momentum flow meters, 204–205, 205f
  categories, 202
  others, 205–206
  pitot tubes, 202, 203, 203f
  rotameter, 205, 205f
  Venturi tube, 202, 202f
Flow variable, in BG model, 362, 362t
Flow velocity meters, 202
Fluorescence-based bio-sensing, 241
Force ripples, 107, 108f
Force sensor for protein delivery, bio-MEMS, 243, 243f
Force transducers, piezoelectric, 193–195, 193f, 194f, 195f
Force vector, in global axes frame, 25
Fourier, Joseph, 139

Fourier analysis, 138, 139–150
  applications, schematic representation, 140, 140f
  Emerson CSI 2120-2 Machinery Health Analyzer, 140f
  FT, *see* Fourier transforms
  overview, 138, 139–141
Fourier integrals, pair, 143
Fourier transforms, 141–150
  application, example of, 143–147
    common waveforms and shapes, 144t
    frequency spectrum of square wave, 147f
    results, 146t
    square wave, 144f, 145f, 146f, 147f
    time domain representation of square wave, 146f
  DFT, 148–149
  FFT, 149–150, 149f
  overview, 141–143
  sampling frequency, 147–148, 148f
Fragment antigen binding (FAB) region, 298
Fragment crystallizable region (FC), defined, 298
Frequency ratio
  dimensionless power *vs.*, 457–458, 458f
  harvested power *vs.*, 458
Frequency response function (FRF)
  defined, 161, 163
  estimators, 165–167, 166f, 167f
  measurement, force transducers in, 194
  for parametric HO model, 267, 268f
  transfer function, 161, 163
Friction, in mechatronic control system, 106–107, 107f
Fuel cell system, 467
Full failure, of antibody, 312, 312f, 313f, 314f, 315–316, 315f
Functional magnetic resonance imaging (fMRI), 570
Function block diagram, PLC language, 103

G

GA, *see* Genetic algorithms (GA)
Gallium arsenide cells (GaAs), 470
Gearbox, IIB conveying system, 387–389, 392f
Genetic algorithms (GA), 308–311
  design evolution, 371
  fitness calculation, 539
  flowchart, 326f
  GP *vs.*, 327
  hybrid GA with GP, 339–341, 340f
  integration with RL, 540–541, 541f
  for mechatronic design, 7
  multi-robot transportation system, 538–540, 538f
  niching, 345–346
  operators, 309, 309f
  optimizing binding affinity function using, 310–311, 311t
  overview, 308–309
  simple, 310
  steps, 326, 327
  sweeping action, 538, 539
Genetically encoded calcium indicators (GECI), 570
Genetic programming (GP)
  BGs and, integration, 328–334
    add-element construction functions, 329, 329f, 330
    arithmetic construction functions, 330
    construction functions and terminals, 329–330
    eigenvalue assignment, case study, 333–334, 334f, 334t
    embryo model, 328, 329f
    first generation, 330–331, 331f
    insert-junction construction functions, 330, 330f
    new generations, creation of, 332–333, 332f, 333f
  design evolution using, 371–372
  evolutionary algorithm, 326–327, 326f, 327f
  hybrid GA with, 339–341, 340f
  LG evolution with, 395–398, 396f, 396t, 397f, 398f, 399f
  niching, 346
Gibbs phenomenon, defined, 147
Global positioning system (GPS), 410
Glucose detection, bio-MEMS sensors, 242, 242f
GP, *see* Genetic programming (GP)
Grabbing mechanism, in UAV cargo transportation system, 438–440, 439f, 440f
Gray code, defined, 210
GremLion, unmanned rotorcraft system
  experimental results, 434, 435f, 436
  hardware system, 424–431
    communication unit, 429
    computers, 427, 428
    control hub, 429
    integration, 429, 431, 431f
    navigation sensors, 427, 428f, 428t
    operating principle, 425, 426f, 427
    servo controller, 428–429
    specifications, 425t

- software system, 432–434
  - ground control, 433, 434, 435f
  - onboard real-time, 432–433, 432f, 433f, 434f

Ground control software system
- coaxial rotorcraft system, 433, 434, 435f
- in unmanned aircraft systems, 424

Gumstix Overo Fire, 415

Gyrators, in LGs, 369, 369f

Gyros, MEMS, 188–189, 188f

H

HAM, *see* Human adaptive mechatronics (HAM)

Hamilton's principle, 474

Hamming window, 159

Hanning window, 159, 159f

Hard disk drive (HDD)
- frequency response, 88, 89f
- physical model of, 88–89, 89f
- state space model of, 90
- system identification approach of, 88

Hardware, unmanned system, 407–421
- actuator management, 415–416
- coaxial rotorcraft system, 424–431
  - communication unit, 429
  - computers, 427, 428
  - control hub, 429
  - integration, 429, 431, 431f
  - navigation sensors, 427, 428f, 428t
  - overview, 424–427, 425f, 425t, 426f
  - servo controller, 428–429
- communication units, 416–417, 416f, 417t
- computers, 415
- integration, 417–421
  - antivibration design, 421
  - computer-aided virtual design, 417, 418f, 419f
  - EMI shielding design, 421
  - power supply design, 421
  - thermal analysis, 420f, 421
- sensors and measurement systems, 407–415
  - GPS/DGPS, 410
  - inertial, 407, 409–410, 409f, 409t
  - lidar systems, 411–412, 411f, 412t
  - magnetometer, 410
  - RGB-D camera, 413–415
  - vision sensor, 412–413
- UAV cargo transportation system, 436–441
  - computers, 441
  - configuration, 437f
  - grabbing mechanism, 438–440, 439f, 440f
  - onboard avionic system, 438f
  - sensors and measurement systems, 440, 441f

Harmonic signals, 128–134
- in Argand plane, 130–132
  - amplitude, 130, 130f
  - circular vector rotating in, 131, 131f
  - perpendicular orthogonal axes, 131
- defined, 128–130, 129f, 130f
- differentiation of, 132–134, 133f
- overview, 128–129

Harvested power, 464
- *vs.* frequency and damping ratio, 458

Health monitoring system (HMS), for IIB, 380, 382

He-Ne laser vibrometers, 181

Hierarchical clustering, 578

High-frequency noise, in mechatronic control system, 110–111

Hokuyo URG-04LX sensor, 505–506, 505f

Holland, John, 309

Home care robotics, 501, 502, 503f

Homogeneous transformation, 508

Hooke's law
- elastic constants and, 44–45
- elastic region by, 80–81
- shear strain/stress distribution, 70
- strains, calculation, 76–77
- stresses, calculation, 52, 79

HOs, *see* Human operators (HOs)

Human adaptive mechatronics (HAM), 254–293
- classical HO models, 256–259
  - crossover, 257, 258–259
  - layout, 256f
  - quasi-linear, *see* Quasi-linear model
- human-in-loop setup, 255–256, 255f
- manual control to, 254–255
- optimal control theory, identification, 268–292
  - data-Based HO model identification, 289–292, 290f, 291f, 292f
  - linear regulator problem, 269
  - LQG controller without time delay, 269–271
  - motor noise effect, 278–279
  - OCM, 274–289
  - overview, 268–269
- quasi-linear model, identification, 259–268
  - experiment and model identification results, 265–268, 266f, 267f, 268f
  - nonparametric, 259, 262–264, 262f
  - parametric, 259, 264–265
  - signal and spectra, 259–262

Human motion, 470
Human nervous system, 568
Human OCM, 276–278, 278f
Human operators (HOs)
  data-based HO model identification, 289–292, 290f, 291f, 292f
  human-in-loop setup, 255–256
  models, 256–259
    crossover, 257, 258–259
    layout, 256f
    quasi-linear, *see* Quasi-linear model
  OCM for, 274–276, 276f; *see also* Optimal control theory
  overview, 254
Human powered devices, 462
Human-powered systems, 470–471, 471f
  instrumented bicycle, 470–471, 471f
Human rescue robotics, 501
Hybrid biological robot (Hybrot), 580–581, 580f
Hybrid GA with GP, 339–341, 340f
Hybrid visual servo controller, using *Q* learning (case study), 545–564, 545f
  experimental results, 558–564
    large dangerous area, 560–561, 561f–564f
    small dangerous area, 558–559, 559f, 560f
  Pioneer^{TM} DX3 mobile robot, 545–546, 545f
  for robust visual servoing, 554–558
    arbitrator of, 558
    control scheme, 554–555, 554f
    Q-learning controller, 555–558, 555f
  vision-based mobile robot motion control, 546–554
    camera projection model, 548–549, 548f
    control law, 549–550
    control scheme, 546, 546f
    experimental results, 550–554, 551f–553f
    kinematic model, 547–548, 547f
"Hybrot" (hybrid biological robot), 580–581, 580f
Hydraulic energy, 454
Hydraulic engine mount design, case study, 349, 351–356
  BG model
    double-notch hydraulic engine mount, 351, 352f
    elite solutions, 352, 353f–354f, 355
    embryo model, 351, 352f
    typical single pumper, 349, 350f
  dynamic stiffness, 351, 351f, 352, 355, 355f, 356f
Hysteresis, 108–109, 108f

I

ICSI, *see* Intra-cytoplasmic sperm injection (ICSI)
Idiotypic network theory
  defined, 299
  of Jerne, 299, 299f
  modified, multi-robot cooperation and, 306–308, 307f, 308f
IEPE (integrated electronic piezoelectric) accelerometer, 177, 179, 179f, 180f
IG-500N, navigation sensor, 427, 428f, 428t
IIB, *see* Intelligent iron butcher (IIB)
Image-based eye-in-hand visual servo control law, 549
Image-based visual servoing (IBVS), 500–501, 501f, 507, 514; *see also* Visual servoing
  traditional, 517–518, 519f–521f
  wheeled mobile robot motion control, 546, 546f
Image plane frame
  of mobile robot, 514
  of robot, 548, 548f
Immune network theory, *see* Idiotypic network theory
Impedances, 335, 336
Impulse response function (IRF), 129, 160, 161, 161f
IMRS, *see* Intelligent multi-robot system (IMRS)
Incremental encoders, 208–209, 209f
Index pulse, defined, 209
Indirect flow meters, 202
Inductive tracing self-organizing map (ITSOM), 583
Inertial measurement unit (IMU), 407, 409–410, 409t
Inertial navigation system (INS), 407, 409–410, 409f
Inertial sensors, unmanned system hardware, 407, 409–410, 409f, 409t
Inertia track
  defined, 349
  fluid moving along, 351f
Infinite impulse response (IIR), filters, 168
Infrared sensors, 200–201, 201f
Innovation representation, defined, 285
Insert-junction construction functions, 330, 330f
Instigating antibody, partial and full failures, 312, 313f, 314f, 315–316
Instigating robot, defined, 301
Instron hydraulic test machine, 195, 195f
Instruction list, PLC language, 103, 103f
Instrumentation, 9–10
Instrumented bicycle, power produced by, 470–471, 471f

Integral causality, in BG modeling, 363–364
Integral gain, defined, 95
Integral windup, defined, 109
Integrated electronic piezoelectric (IEPE) accelerometer, 177, 179, 179f, 180f
Integration, hardware
  unmanned aircraft systems, 417–421
    antivibration design, 421
    coaxial rotorcraft system, 429, 431, 431f
    computer-aided virtual design, 417, 418f, 419f
    EMI shielding design, 421
    power supply design, 421
    thermal analysis, 420f, 421
Integration, of BGs and GP, 328–334
  BG embryo model, 328, 329f
  construction functions and terminals, 329–330
    add-element functions, 329, 329f, 330
    arithmetic functions, 330
    insert-junction functions, 330, 330f
  eigenvalue assignment, case study, 333–334, 334f, 334t
  first generation, 330–331, 331f
  new generations, creation of, 332–333, 332f, 333f
Integrator windup, 99–100, 99f, 100f
Intelligent iron butcher (IIB), 377, 379–400
  design evolution algorithm, 380–382, 381f
  illustrative scenario, 382–386, 395–400
    block diagram of hydraulic system, 382f
    design improvement of conveying system, 398, 400f
    fitness function, 383–384, 383f
    LG evolution with GP, 395–398, 396f, 396t, 397f, 398f, 399f
    response curve, 382f
    servo valve and hydraulic actuator, 383, 383f
    system identification, 384–386, 385f, 386f
  LG methodology, conveying system, 386–394
    electromechanical, 387–389, 388f
    gearbox, 387–389, 392f
    lever, 387–389
    push–pull movement, 393f
    SimMechanics model, 391, 393, 394f
    sliding mechanism, 387f
    state-space model, 387–391
    VDP drive, 386, 387, 391
    VFD, 391, 391f
    wheel/axle, 387–389
  operation, 379
  overview, 377, 379, 379f
  performance and condition, monitoring, 380
Intelligent multi-robot system (IMRS), 296–318
  AIS, 300–301
    multi-robot cooperation and, 305–308
    network theory model, 300–301
  biological immune system, 297–299
  GA, 308–311
    operators, 309, 309f
    optimizing binding affinity function using, 310–311, 311t
    overview, 308–309
    simple, 310
  Jerne’s idiotypic network theory, 299, 299f
  multi-robot cooperation problem, 301–304
    decision conflicts, 302–303
    fault tolerance, 302
    interdependencies and priorities, 303–304, 304f
  overview, 296–297
  results and discussion, 311–318
    antibody failures, 312, 312f, 313f, 314f, 315f
    assumptions, on developed methodology, 311
    eliminating antigen by single antibody, 317, 317f, 318f
    selecting appropriate antibodies, 311–312
Intensity quantities, 211, 212–213
Interburst event interval (IBEI) histogram, 577
Interburst interval (IBI), 573
Interdependencies, multi-robot cooperation problem, 303–304, 304f
Internal electrical energy, 475
Internal model control (IMC) scheme, 113, 113f
Interspike interval (ISI), 573
Intra-cytoplasmic sperm injection (ICSI), 116–122
  adaptive control, 122, 122f
  installation, 118, 119f
  linear reciprocating motion, 117
  LVDT, 117–118, 118f
  oocytes, structure of, 116–117, 116f
  PID control, 119–121, 120f
  Simulink®, 118, 119f
  system identification, 119
In vitro microelectrode arrays (MEAs)
  array-wide bursting, 574
  commercial, 572–573
  dynamics of recordings, 573–575
    bursts, 573, 574f
    network bursts, 574–575, 574f
    spikes, 573
  embodied neuronal networks, 578–584
    supervised learning, 579–581
    unsupervised learning, 581–584

  "firing," 571
  functionality, 570–571, 571f
  local field potentials (LFP) analysis, 573
  multi-unit recordings, 575
  network dynamics, detection of
    burst detection, 576, 576f
    functional motifs identification, 577–578
    general analysis methods, 577, 578f
    network burst detection, 577
    spike detection, 575
    spike sorting, 575
  NeuroExplorer, 573
  and other neural recording techniques, 570
    optical methods, 570
  single-well, 571f
  strengths and weaknesses, 571–572
  systems and software, 572–573, 572f
Ion conductive polymer film (ICPF) micropump, 239, 239f
Ionic EAP, 223–225, 223f, 224f
Ionic polymeric-conductor composites (IPCCs), 223, 223f
Ionic polymeric-metal composites (IPMCs), 223–225, 224f
iRobot-PackBot EOD, 503f, 504
Iron butcher
  controller design, case study, 341–345, 341f, 342f, 343f, 344f, 345f
  IIB, *see* Intelligent iron butcher (IIB)
Irradiance absolute value, 468–469
ISO 20473:2007 standard, 201
ITSOM (inductive tracing self-organizing map), 583

J

Jellyfish locomotion
  self-propelled vehicle inspired by, 478–481, 479f–480f
    pulsed-jet propulsion, 479, 480–481, 480f
    steady-jet propulsion, 479–480, 480f
Jerne, Neil, 299
Jerne's idiotypic network theory, 299, 299f
Junctions, in BG model, 361

K

Killer T cells, defined, 297
Kinematic modeling, of robots, 508–509, 509t, 510f
  coordinate frames, 509, 510f
  Denavit-Hartenberg (DH) convention, 508–509, 509t
  homogeneous transformation, 508
  vision-based mobile robot motion control, 547–548, 547f
Kinetic energy, 454
  human motion and, 470
  as source for energy harvesting, 455, 456
  of walking, 470
Kronecker delta, 460
K-type thermocouple, 199, 199f
KUKA OmniRob, 503f, 504

L

LabVIEW™, 167
Ladder diagram, PLC language, 103, 103f
Lagrange equation, 475
Lambda
  power *vs.*, 466–467, 466f
Large dangerous area, hybrid visual servoing with, 560–561, 561f–564f
Laser distance finders, 505–506
  Hokuyo URG-04LX, 505–506, 505f
  SICK LMS 200 2-D scanner, 505f, 506
Laser Doppler velocimeters (LDVs), 181–183, 182f, 205
Lasers, for measurement of displacement, 186–187, 186f
"Leader neurons," 577
Leakage, 155–160
  defined, 156, 157f
  examples, 156, 156f
Learning ability, multi-robot system, 536
Learning and evolution agent, multi-robot transportation system, 536
Lidar systems, 411–412, 411f, 412t
Linear actuator, 461
Linear continuous time filters, 168
Linear ensemble averaging, 167
Linear frequency plots, logarithmic *vs.*, 214, 214f
Linear graphs (LGs)
  evolution with GP, 395–398, 396f, 396t, 397f, 398f, 399f
  methodology, IIB conveying system, 386–394
    electromechanical, 387–389, 388f
    gearbox, 387–389, 392f
    lever, 387–389
    push–pull movement, 393f
    SimMechanics model, 391, 393, 394f
    sliding mechanism, 387f
    state-space model, 387–391
    VDP drive, 386, 387, 391
    VFD, 391, 391f
    wheel/axle, 387–389

  modeling, 368–371
    branch of element, 368, 368f
    compatibility equations, 369, 371
    constitutive equations, 369, 370f, 371
    continuity equations, 369, 371
    elements, types, 368–369, 369f
    representations of simple mechanical system, 369, 370f
    transformers and gyrators, 369, 369f
    variables, through and across, 368, 368t
Linear model, equations of motion, 19
Linear momentum force systems, 19–20
  generalization of equation, 20
  stiffness and damping systems, 19–20
Linear quadratic Gaussian (LQG) controller
  defined, 269
  with time delay, 271–274, 274f
  without time delay, 269–271
Linear regulator problem, 269
Linear variable (voltage) differential transformers (LVDTs), 117–118, 118f, 184–185, 185f
Load cells, 193–198
  pair force transducer *vs.* accelerometer, calibration, 197–198, 197f
  piezoelectric force transducers, 193–195, 193f, 194f, 195f
  strain gauge–based, 195–196, 195f, 196f
Loads
  defined, 39
  freely supported beams and
    asymmetrical, 48–50, 48f
    symmetrical, 47–48, 47f
    uniform, 50–51, 50f
Local field potentials (LFP) analysis, 573
Logarithmic frequency plots, linear *vs.*, 214, 214f
Logarithmic scales, 211–217
  decibel (dB), 211–215
    dB power *vs.* root-power ratio scales, 214, 215t
    linear *vs.* logarithmic frequency plots, 214, 214f
    power (intensity) quantities, 211, 212–213
    reference values, 214, 215t
    root-power (amplitude) quantities, 211, 213–214
    sound-level meter, 212f
  octaves, 215–217
    bandwidth of, 216–217, 216f, 217f
    defined, 215
Low-frequency drift, in mechatronic control system, 110
Low-frequency energy harvester, 454
Low-pass filters, 168
Lyapunov method, 518, 550
Lymphocytes, 297

M

Macaulay's method, 65–69, 65f, 67f, 68f, 69f
Machine learning
  multi-robot transportation using (case study), 534–545, 535f
    cooperation strategy, 536–541
    evolutionary learning mechanism, 541–543, 542f–543f
    experimentation, 543–545, 544f
    genetic algorithms, 538–540, 538f
    multi-agent infrastructure, 534–536, 535f
    objective of, 535
    reinforcement learning, 536–538, 537f
    reinforcement learning and genetic algorithms, integration scheme, 540–541, 541f
    simulation results, 541–543, 542f–543f, 544f
    sweeping action, 543, 544f
  overview, 530
Machine vision, 499
Mach–Zehnder interferometer, 181, 182f
Magnetic tweezers (MT), for micromanipulation of biomolecules, 230, 231f
Magnetoencephalography (MEG), 570
Magnetometer, 410
Markov decision process (MDP), 530–534
  attributes, 531
  Bellman equation, 532
  defined, 530
  "utility of a state," 531–532
  value iteration algorithm, 531, 532, 532f
Mass-spring-damper system
  with base excitation, 455f
  dimensionless power, 457
    vs. frequency ratio and damping ratio, 457–458, 458f
  equation of motion of, 455–456
  harvested power *vs.* frequency and damping ratio, 458
  kinetic energy of, 456
  power calculation, 456–458
Matching index, 375, 383–384
Materials, mechanics, *see* Mechanics, of materials
MATLAB®, 118, 266, 267, 385, 391
MATLAB Compiler™, 523
MATLAB MPC Toolbox™, 523

MBDS, software, 32–34
  front screen, 32f
  response of system to simple step function, 33, 34, 34f
  two-mass spring system with actuator and relative velocity sensor, 33, 33f
MC_Rack's burst detection algorithm, 576
MDP, *see* Markov decision process (MDP)
MeaBench, 573, 577
MEATools, 573
Mechanical micropumps, 235
Mechanics, of materials, 38–83
  beams, deflection of, 55–69
    bending moments, 57, 58f
    discontinuous bending moment equations, 64, 64f
    equilibrium and determinacy, 57, 57f, 58f
    flexure equation, 56–57, 56f, 58, 59
    overview, 55–56
    singularity function (Macaulay's) method, 65–69, 65f, 67f, 68f, 69f
    statically indeterminate beams, 62–64, 63f
    transverse loaded slender, 59–62, 60f, 61f
  bending, theory, 45–55; *see also* Bending, theory
    defined, 46
    moment and shearing force, 46–51
    overview, 45–46
    planar bending model, 46, 46f
  deformation, 42
  elasticity, 38–45
    Hooke's law and elastic constants, 44–45
    modulus of, 45
    stress and strain, 38–43
    yield and, 44
  load, 39
  properties, 80–83
    overview, 80
    Poisson's ratio, 82–83, 82f
    stress–strain behavior of ductile materials, 80, 81, 81f
    tension and compression tests, 80, 80f, 81f
  strains
    analysis, 77–79
    compressive, 43
    direct, 42–43
    gage rosettes, 77–79
    overview, 38–39
    principal strains to principal stresses, conversion, 79
    shear, 43
    tensile, 42
    volumetric, 43–44, 43f
  stresses, 38–45; *see also* Stresses
    complementary shear, 41–42, 42f
    compressive, 39, 40–41, 40f
    defined, 39
    direct/normal, 39–41
    nonuniform, 41
    overview, 38–39
    principal strains to principal stresses, conversion, 79
    shear, 39, 41, 41f
    tensile, 39–40, 40f
    transformation in two dimensions, 72–77
  torsion, theory, 69–72
    overview, 69
    rate of twist, 70, 71, 71f
    shear strain/stress distribution, 70, 70f, 71f
    shear stress from, 72
Mechanobiology, defined, 233–234
Mechatronic design quotient (MDQ), 7–8, 375
Mechatronic engineering, 1–12
  application areas, 11–12
  designs, 3–9
    coupled, 5–7, 6f
    evolution, 8–9, 9f
    GA, 7
    index, 6
    MDQ, 7–8
    modeling and, 3–4, 4f
    problems, 5
  evolution, 10
  hard-disk drive, 2, 2f
  instrumentation, 9–10
  modeling, 3–4, 4f
  overview, 1–3, 2f, 3f
  technology issues, 2, 3f
Mechatronics; *see also* Neuromechatronics
  defined, 568
  evolution of, 568
Mechatronics design, of unmanned aircraft systems, *see* Unmanned aircraft systems
Mechatronic systems, control
  application examples, 114–122
    flight simulators, 114–116, 115f
    piezoelectric control system for biomedical application, 116–122, 116f, 118f, 119f, 120f, 122f
  challenges, 106–113
    dead zone model, 109, 110f
    force ripples, 107, 108f
    friction, 106–107, 107f
    high-frequency noise, 110–111

    - hysteresis and backlash, 108–109, 108f
    - low-frequency drift, 110
    - nonlinear dynamics, incorporating and addressing, 111–113, 113f
    - reference signal changes, 109, 110
    - saturation model, 109, 109f
  - computer control, implementation, 103–106, 104f, 105f
  - control systems, 88–93
    - components, 86
    - defined, 86
    - performance assessment, 92–93, 92f
    - system model, 88–92, 89f, 91f, 92f
  - example of, 86
  - historical perspective, 86, 87f
  - structure, 86, 87f
  - techniques, 93–103
    - feedforward control, 101, 101f
    - PID control, 93–100; *see also* Proportional-integral-derivative (PID) control
    - PLC, 102–103, 103f
    - servo control structure, 101–102, 102f
- Mechatronic systems, design evolution of, *see* Design evolution
- Methodology/methods
  - Barlett's method, 167
  - collocated control, 335–336, 335f, 336f
  - design evolution, 374
    - application, to industrial systems, 377–400
    - LGs, application of, 386–394
  - Macaulay's method, 65–69, 65f, 67f, 68f, 69f
  - pole placement, 343
  - prediction error, 265
  - priority-based method, robots and, 303–304, 304f
  - singularity function, 65–69, 65f, 67f, 68f, 69f
  - spectral estimation, 267, 267f
  - Welch's method, 167
  - Ziegler–Nichols method, 98–99
- Micro electromechanical system (MEMS) sensors
  - bio-MEMS, *see* Biomedical MEMS (bio-MEMS)
  - overview, 188–189, 188f
- Microfluidic devices, capillary valves in, 233–235, 234f
- Micro-nano-electromechanical systems (MNEMS) tweezers, 231, 231f, 232f
- Micropumps
  - bimetallic, 238, 239, 239f
  - electrostatic, 236, 237f
  - ICPF, 239, 239f
  - mechanical, 235
  - nonmechanical, 235
  - piezoelectric, 237
  - SMA, 238, 238f
  - thermo-pneumatic, 238
- Microsoft Kinect, 414, 506, 506f
- Mid-infrared (MIR) region, 201
- Minimally invasive surgery (MIS), 243, 244, 244f
- Mobile robotic visual servo systems, 499–526; *see also specific types*
  - adaptive nonlinear model predictive control, 518, 522–525, 522f, 524f–526f
  - advantages, 502
  - block diagram of, 500, 500f
  - camera configurations, 504–505, 504f
  - camera modeling, 510–513, 511f–512f
  - camera parameters, 513–514
  - DARPA Urban Challenge, 501–502, 502f
  - dynamic look-and-move structure, 500, 500f
  - kinematic modeling, 508–509, 509t, 510f
  - object-grasping task by, 514, 515f
  - overview, 499–501
  - sensors, 504–507, 504f–507f
    - camera configurations, 504–505, 504f
    - laser distance finders, 505–506, 505f
    - sonar, 505–506
    - stereo vision, 506–507, 506f–507
  - state of the art of, 501–504
  - system modeling, 514–517, 515f–516f
    - coordinate frames, 514, 516, 516f
  - traditional image-based system, 517–518, 519f–521f
- Modeling, mechatronic systems, 361–371
  - bond graph (BG) modeling, 361–368
    - causality assignment, 363–364, 364f, 364t
    - electrical and mechanical system, 362, 363, 363f
    - elements, 361, 362f
    - state-space model, 365–368, 365t, 366f
    - variables, 362, 362t
  - linear graphs (LGs), 368–371
    - branch of element, 368, 368f
    - compatibility equations, 369, 371
    - constitutive equations, 369, 370f, 371
    - continuity equations, 369, 371
    - elements, types, 368–369, 369f
    - representations of simple mechanical system, 369, 370f
    - transformers and gyrators, 369, 369f
    - variables, through and across, 368, 368t
- Modeling/models
  - for control of rigid bodies in 3-D space, 26–32
    - actuators, 26–30
    - sensors, 18, 26–27, 30–32

  - dead zone model, 109, 110f
  - design and, 3–4, 4f
  - equations of motion for linear model, 19
  - hysteresis and backlash model, 108–109, 108f
  - intelligent iron butcher (IIB) conveying system, 386–394
    - electromechanical, 387–389, 388f
    - gearbox, 387–389, 392f
    - lever, 387–389
    - push–pull movement, 393f
    - SimMechanics model, 391, 393, 394f
    - sliding mechanism, 387f
    - state-space model, 387–391
    - VDP drive, 386, 387, 391
    - VFD, 391, 391f
    - wheel/axle, 387–389
  - planar bending model, 46, 46f
  - quarter car model, 33
  - saturation model, 109, 109f
  - system model, 88–92
    - actuators, linear dynamics, 91
    - electric representation of DC motor system, 91, 92f
    - HDD, 88–90, 89f
    - thermocouple schematic, 90, 91f
- Model-referenced active car suspension, case study, 347–349, 347f, 349t, 350f
- Modified idiotypic network model, multi-robot cooperation and, 306–308, 307f, 308f
- Modified optimal control model (MOCM), 279–283, 280f
- Modulus of elasticity, defined, 45
- Moment of inertia
  - area, 53, 54–55, 54f
  - equivalent
    - of gear, 389
    - of wheel, 389
  - polar, 71, 72
- Moment of momentum, equations of, 21
- Mori, Tetsura, 86
- Motifs
  - identification, in MEA recordings, 577–578
- Motion, equations of
  - assembly, 21–26, 22f
  - for linear model, 19
- Motion control, of vision-based mobile robot, 546–554
  - camera projection model, 548–549, 548f
  - control law, 549–550
  - control scheme, 546, 546f
  - experimental results, 550–554, 551f–553f
  - kinematic model, 547–548, 547f
  - visual errors, 546
- Motor constant, 464
- Motor noise effect, 278–279
- Motor torque constant, 461
- Motor voltage constant, 461
- Moving average, 174–175, 175f
- Multi-agent infrastructure, of multi-robot transportation, 534–536, 535f
  - learning and evolution agent, 536
  - physical agents, 535–536
  - software agents, 535
  - vision agent, 536
- Multibody Advance Airship for Transport (MAAT), 466–467, 467f
  - buoyancy force, 467
  - external forces on, 467–468, 468f
  - power system, 468–470, 469f–470f
- Multidomain engineering systems, *see* Mechatronic systems
- Multi-electrode array art (MEART), 581
- Multielectrode arrays (MEAs), *see In vitro* microelectrode arrays (MEAs)
- Multiple-input-multiple-output (MIMO) system
  - FRF for harmonic excitation on, 167, 167f
- Multi-robot cooperation
  - AIS and, 305–308
    - binding affinity, 305
    - modified idiotypic network model, 306–308, 307f, 308f
    - robot and antibody, 306, 306f
  - problem, 301–304
    - decision conflicts, 302–303
    - fault tolerance, 302
    - interdependencies and priorities, 303–304, 304f
- Multi-robot transportation, using machine learning (case study), 534–545, 535f
  - cooperation strategy, 536–541
    - genetic algorithms, 538–540, 538f
    - reinforcement learning, 536–538, 537f
    - reinforcement learning and genetic algorithms, integration scheme, 540–541, 541f
  - evolutionary learning mechanism, 541–543, 542f–543f
  - experimentation, 543–545, 544f
  - learning ability, 536
  - multi-agent infrastructure, 534–536, 535f
  - objective of, 535
  - simulation results, 541–543, 542f–543f, 544f
  - sweeping action, 543, 544f
- Multi-unit recordings, 575
- Mutation, operation of GP, 309, 332, 376
- MvBlueFOX camera, 440

N

Nanotweezers, for micromanipulation of biomolecules, 229–233
  AFM, 229, 229f
  MNEMS, 231, 231f, 232f
  MT, 230, 231f
  multiscale actuation mechanism, 232–233, 233f
  OT, 230, 230f
NASA/GM Robonaut, 503f, 504
NASA Mars Rover, 503f, 504
National Instruments, 147
Navigation sensors, coaxial rotorcraft system, 427, 428f, 428t
Near infrared (NIR) region, 201
Neobotix, 503f, 504
Nervous system, 568
Network bursts, in MEA recordings, 574–575, 574f
  detection of, 577
Network theory model, 300–301
Neural–robotic interface
  ANNs for, 583–584, 584f
NeuroExplorer, 573
NeuroExplorer's burst detection algorithm, 576
Neuromechatronics, 568–569
  defined, 568
  with *in vitro* microelectrode arrays, *see In vitro* microelectrode arrays (MEAs)
Neuronal networks, 569–570
  cultured, 571f
  embodied, 578–584
    supervised learning, 579–581
    unsupervised learning, 581–584
  network dynamics, detection of
    burst detection, 576, 576f
    functional motifs identification, 577–578
    general analysis methods, 577, 578f
    network burst detection, 577
    spike detection, 575
    spike sorting, 575
Neurons
  clustering of, 578
Neutral axis, location of, 54, 54f
Newton's second law, 492
Niching optimization scheme, 345–356
  GA, 345–346
  GP, 346
  hydraulic engine mount design, case study, 349, 351–356; *see also* Hydraulic engine mount design
  model-referenced active car suspension, case study, 347–349, 347f, 349t, 350f
Nondeterministic signals, 128
Nonlinear dynamics, in mechatronic control system, 111–113, 113f
Nonlinear model predictive control (NMPC), 518, 522
  adaptive, 518, 522–525, 522f, 524f–526f
Nonmechanical micropumps, 235
Nonparametric quasi-linear model, 259, 262–264, 262f
Nonperiodic signals, 128–129
Nonstationary signals, 128
Nonuniform stress, 41
Normalized power
  regenerative system, 465–466, 465f
Notch frequency, defined, 351
NUS^{2}T-Lion
  experimental results, 445–447, 446f, 447f
  hardware system, 436–441
    computers, 441
    configuration, 437f
    grabbing mechanism, 438–440, 439f, 440f
    onboard avionic system, 438f
    sensors and measurement systems, 440, 441f
  overview, 436, 436f
  software system, 442–445, 443f, 444f
Nyquist–Shannon theorem, 150, 152

O

Object-grasping task, by mobile robot, 514, 515f
Octaves, 215–217
  bandwidth of, 216–217, 216f, 217f
  defined, 215
Offline Spike Sorter software, 573
Offset sensor configuration, for incremental encoders, 208–209, 209f
Offset track configuration, for incremental encoders, 209, 209f
Onboard real-time software system
  coaxial rotorcraft system, 432–433, 432f, 433f, 434f
  unmanned aircraft system, 421, 423–424
One-port active/passive elements, in BG model, 361
Online monitoring, 8
Oocytes, structure of, 116–117, 116f
Oolemma, 116–117, 116f
OpenCV, 414, 415
Operation modes, regenerative system, 464
Operators, GA, 309, 309f
Optical imaging techniques, 570

Optical tweezers (OT), for micromanipulation of biomolecules, 230, 230f
Optics, 570
Optimal bounds, on uncertainties, 492
Optimal control theory, identification, 268–292
  data-Based HO model identification, 289–292, 290f, 291f, 292f
  linear regulator problem, 269
  LQG controller without time delay, 269–271
  model (OCM)
    for HO, 274–276, 276f
    human, 276–278, 278f
    identification, 283–289
    MOCM, 279–283, 280f
  motor noise effect, 278–279
  overview, 268–269
Optimal uncertainty quantification (OUQ), 492
  certification problem in, 493
Optogenetics, 570
Ordinary coherence function, defined, 166
Orifice flow meters, 202
Oscillating mass-spring system, 130
Overlapping, 167

P

Pade approximation, 279, 280
Pair force transducer *vs.* accelerometer, calibration, 197–198, 197f
Pan-tilt-zoom (PTZ) vision system, 546
Parameter variation index (PI), defined, 384
Parametric quasi-linear model, 259, 264–265
Paratope
  of antibody, 298–299, 300–301, 305, 306–307, 307f, 308f
  defined, 298
Partial failure, of antibody, 312, 312f, 313f, 314f, 315–316, 315f
Passive sensors, 177
Pathogens, defined, 297
Patterned training stimuli (PTS), 583
PC-104–based flight control computer, 415
Peak amplitude, 130, 130f
Performance assessment, of control system, 92–93, 92f
Periodic signals, 128–129
Peripheral nervous system (PNS), 568
Phagocytes, 297
Photovoltaic (PV) cells, 467
  current–voltage relationship of, 469–470, 470f
  electrical power system of, 469
Physical agents, multi-robot transportation system, 535–536
Pico-Coloumb (pC), 178, 179
PID, *see* Proportional-integral-derivative (PID) control
Piezoelectric accelerometers, 177, 178–180, 178f, 179f, 180f
Piezoelectric actuators, 237
Piezoelectric cantilever, 472–473, 472f–473f
Piezoelectric constant, 460
Piezoelectric control system, for biomedical application, 116–122
  ICSI installation, design of, 116–122, 116f
    adaptive control, 122, 122f
    linear reciprocating motion, 117
    LVDT, 117–118, 118f
    oocytes, structure of, 116–117, 116f
    PID control, 119–121, 120f
    Simulink®, 118, 119f
    system identification, 119
Piezoelectric energy
  conversion mechanism, 459–461, 460f
Piezoelectric energy harvester, 455
  bio-inspired, 472–473, 472f–473f
Piezoelectric energy harvesting
  aeroelastic vibration for, 473–477, 474f
Piezoelectric force transducers, 193–195, 193f, 194f, 195f
Piezoelectricity, linearized theory of, 460
Piezoelectric micropumps, 237
Piezoresistive accelerometer, 180–181, 181f
Pinhole camera model, 510–511, 511f
Pioneer™ DX3 mobile robot, 545–546, 545f
Pitot tubes, 202, 203, 203f
Pixel coordinate frame
  of mobile robot, 514
  of robot, 548, 548f
Planes of maximum shear, defined, 75
Plane stress, in two dimensions, 74–76, 74f
Plasticity
  training methods and, in unsupervised learning, 582–583, 582f
PLC (programmable logic controller), 102–103, 103f
Plexon, 573
PMDTec, 414
Poisson's ratio, property of materials, 82–83, 82f
Polarizing beam splitter (PBS), 181–182
Pole placement method, 343
Poly acrylamide-ran-3-acrylamidophenylboronic acid (PAA-ran-PAAPBA), 242
Polycrystalline cells, 470

Polydimethylsiloxane (PDMS) nanorods, 226, 228f, 229
Polyimides, 226, 227
Polymers, in artificial muscles, *see* Electroactive polymers (EAP)
Polyvinylidene fluoride (PVDF), 244
Position-based visual servoing, 500–501, 501f, 507; *see also* Visual servoing
Power
  normalized, 465–466, 465f
  produced by instrumented bicycle, 470–471, 471f
  regenerative system, 465–466, 465f
  vs. lambda and velocity, 466–467, 466f
Power calculation
  mass-spring-damper system, 456–458
Power moderation, in flight simulators, 114
Power quantities, 211, 212–213
Power-spectral density (PSD), 162–163, 163f, 164, 164f
Power supply design, in unmanned aircraft systems, 421
Power system
  airships, 468–470, 469f–470f
  PV cells, 468–470, 470f
Prandtl tube, 203, 203f
Precision timing spike detection (PTSD), 575
Prediction error method, defined, 265
Pressure transducers, 206–207, 206t, 207f
PrimeSense, 414
Principal plane, defined, 75
Principal strains, principal stresses to, 79
Principal stresses
  defined, 75
  principal strains to, conversion, 79
Principle component analysis (PCA), 578
Priority-based method, robots and, 303–304, 304f
Probability distribution
  defined, 531
Programmable logic controller (PLC), 102–103, 103f
Properties of materials, mechanical, 80–83
  overview, 80
  Poisson's ratio, 82–83, 82f
  stress-strain behavior of ductile materials, 80, 81, 81f
  tension and compression tests, 80, 80f, 81f
Proportional-integral-derivative (PID) control, 93–100
  constituent components, 93
  ICSI system and, 119–121, 120f
  integrator windup, 99–100, 99f, 100f
  performance, 93–97, 94f, 96f, 97f
  robustness, 98
  system response speed and stability, 98t
  Ziegler-Nichols method, 98–99
Proportional-integral-derivative (PID) controller, of robot, 546
Protein delivery, MEMS force sensor for, 243, 243f
Proximity probes
  displacement transducer, 187–188, 187f
  as tachometer, 183–184, 184f
PSD (power-spectral density), 162–163, 163f, 164, 164f
Pulsed-jet propulsion
  bio-inspired self-propelled vehicle, 479, 480–481, 480f
Pursuit tacking system, 256
Push rods, 194, 194f
Pyrometers, 187, 201

Q

Q learning, 530–534, 532f, 533f; *see also* Reinforcement learning
  ε – *greedy* search policy, 533–534
  hybrid visual servo controller using (case study), 545–564, 545f; *see also* Hybrid visual servo controller
    experimental results, 558–564
    Pioneer^{TM} DX3 mobile robot, 545–546, 545f
    for robust visual servoing, 554–558
    vision-based mobile robot motion control, 546–554
  Q table, 533
  single-agent algorithm, 533, 533f
Q-learning controller, 555–558, 555f, 559–560
  advantage of, 555
  states, actions, and rewards, 555–557, 555f
  training results, 557–558, 557f
QNX Neutrino, 424
Q table, 533
Quadratic programming algorithm (QP), 523
Quantification, of signals, 134–138, 135f, 136f, 137f
Quantization, errors, 155, 155f
Quarter car model, defined, 33
Quasi-linear model
  HO model, 257, 257f
  identification, 259–268
    experiment and model identification results, 265–268, 266f, 267f, 268f
    nonparametric, 259, 262–264, 262f

    parametric, 259, 264–265
    signal and spectra, 259–262

R

Random signals, 128, 161–167
  ASD, 162–163
  CSD, 164–165
  ensemble averaging, 167
  estimators, 165–167, 166f, 167f
  PSD, 162–163, 163f, 164, 164f
  spectral analyzers, 165, 165f
Raptor 90 helicopter, 405, 438
Rate of twist, torque and, 70, 71, 71f
Real-time computing, 499
Real-time kinematic (RTK) navigation method, 410
Real-time spike sorting, 575
Recordings, MEA; *see also In vitro* microelectrode arrays (MEAs)
  dynamics of, 573–575
    bursts, 573, 574f
    network bursts, 574–575, 574f
    spikes, 573
Rectangular window, 159
Reference signal changes, 109, 110
Reference values, decibel, 214, 215t
Reflected infrared region, 201
Regenerative actuators, 454
  electromagnetic, 464
Regenerative system
  brake mode, 464
  drive mode, 464
  dynamic characteristics, 465–466
  dynamic force, 466
  with electromagnetic energy conversion, 463–464, 463f; *see also* Self-powered dynamic system
  motor constant, 464
  normalized power, 465–466, 465f
  operation modes, 464
  power, 465–466, 465f
    vs. lambda and velocity, 466–467, 466f
  regeneration mode, 464
Reinforcement learning (RL); *see also Q* learning
  development, 532
  integration with GA, 540–541, 541f
  multi-robot transportation system, 536–538, 537f
  overview, 532
  variants of, 532–533
Reluctance force, 107
Remnant statistics, HO and, 257
Remote sensing, 454, 462
Renewable energy
  for dynamic systems, 466–470, 467f, 468f–470f
  human motion and, 470
Repetition, genetic operation, 376
Reset windup, 99–100, 99f, 100f
Resistance temperature detectors (RTDs), 199–200
Resistance thermometers, 199–200
Rewards, *Q*-learning controller, 555–557, 555f
RGB-D camera, 413–415
Riccati equation, 269, 270, 271, 277, 287
Rigid bodies in 3-D space, control of, 17–34
  MBDS, software, 32–34
    front screen, 32f
    system to simple step function, response, 33, 34, 34f
    two-mass spring system, 33, 33f
  modeling for, 26–32
    actuators, 18, 26–30
    sensors, 18, 26–27, 30–32
  overview, 18, 18f
  theory, 19–26
    assembly of equations, 21–26, 22f
    definitions and assumptions, 19
    linear momentum force systems, 19–20
    moment of momentum, equations of, 21
    motion for linear model, equations of, 19
RL, *see* Reinforcement learning (RL)
RMS (root mean square), quantification of signals by, 134–138, 135f, 136f, 137f
Robot frame, of mobile robot, 514, 516f, 547–548, 547f
"Robot-in-the-loop" systems, 568
Robot modeling, 499
Robots
  antibody and, 306, 306f
  IMRS, *see* Intelligent multi-robot system (IMRS)
  multi-robot cooperation; *see also* Multi-robot cooperation
    AIS and, 305–308
    problem, 301–304
Robots/robotic systems; *see also* Mobile robotic visual servo systems
  hybrid visual servo controller using *Q* learning (case study), 545–564, 545f
    experimental results, 558–564
    for robust visual servoing, 554–558
    vision-based mobile robot motion control, 546–554

  - kinematic modeling, 508–509, 509t, 510f
    - coordinate frames, 509, 510f
    - Denavit-Hartenberg (DH) convention, 508–509, 509t
    - homogeneous transformation, 508
  - Markov decision process (MDP), 530–534
  - multi-robot transportation using machine learning (case study), 534–545, 535f
    - cooperation strategy, 536–541
    - evolutionary learning mechanism, 541–543, 542f–543f
    - experimentation, 543–545, 544f
    - genetic algorithms, 538–540, 538f
    - multi-agent infrastructure, 534–536, 535f
    - objective of, 534
    - reinforcement learning, 536–538, 537f
    - reinforcement learning and genetic algorithms, integration scheme, 540–541, 541f
    - simulation results, 541–543, 542f–543f, 544f
    - sweeping action, 543, 544f
  - *Q* learning, 530–534, 532f, 533f

Root mean square (RMS), quantification of signals by, 134–138, 135f, 136f, 137f

Root-power quantities, 211, 213–214

Root-power ratio scales, dB power *vs.*, 214, 215t

Rosettes, 190, 191, 191f

Rotameter, 205, 205f

Rotational variable differential transformer (RVDT), 184

RTLinux, 424

S

Sallen–Key topology, 171

Sampling frequency, 147–148, 148f

Saturation model, 109, 109f

Savitzky–Golay filtering, 175–177, 176f, 176t

Section property, in bending, 53–55
  - centroid, position of, 53f
  - neutral axis, location, 54, 54f
  - second moment of area, 54–55, 54f

Seebeck, Thomas Johann, 199

Seebeck effect, 199

Selection operator, GA, 309, 309f

Self-organizing maps (SOMs), 581–582

Self-powered dynamic system
  - concept of, 462–463, 462f
  - defined, 462
  - energy harvesting techniques, 455–461
    - concept of, 455–456, 455f
    - conversion mechanisms, 459–461, 459f–461f
    - kinetic energy for, 455, 456
    - mass-spring-damper system, 455–456, 455f
    - overview, 454
    - sources, 455
  - human-powered systems, 470–471, 471f
  - overview, 454–455
  - regenerative scheme, 463–464, 463f; *see also* Regenerative system
  - renewable energy for, 466–470, 467f, 468f–470f
  - theory of, 463–466, 463f, 465f–466f
  - uncertainty quantification, 491–492, 493–494

Self-powered regenerative shock absorber, 464

Self-powered sensors, 454, 462
  - self-powered regenerative shock absorber, 464

Self-powered vibration control system, 454

Self-propelled vehicle, bio-inspired, 478–481, 479f–480f
  - pulsed-jet propulsion, 479, 480–481, 480f
  - steady-jet propulsion, 479–480, 480f

Sensors, 177–211
  - accelerometers, 177, 178–181
    - piezoelectric, 177, 178–180, 178f, 179f, 180f
    - piezoresistive and capacitive, 180–181, 181f
  - active and passive, 177
  - analog and digital, 177
  - bio-MEMS, 240–245
    - acoustic, 245
    - blood cell counter, 244, 244f
    - for CRP detection, 241, 242, 242f
    - force sensor for protein delivery, 243, 243f
    - glucose detection, 242, 242f
    - tissue softness characterization, 243, 244, 244f
    - triglyceride biosensor, 241, 241f
  - for control of rigid bodies in 3-D space
    - modeling, 18, 26–27, 30–32
    - velocity, two-mass spring system and, 33, 33f
  - displacement transducers, 184–189
    - lasers, 186–187, 186f
    - LVDTs, 184–185, 185f
    - MEMS sensors, 188–189, 188f
    - proximity probes, 187–188, 187f
  - flow, 201–206
    - anemometers, 203–204, 204f
    - angular momentum flow meters, 204–205, 205f
    - categories, 202
    - others, 205–206

        pitot tubes, 202, 203, 203f
        rotameter, 205, 205f
        Venturi tube, 202, 202f
    generic sensor setup, 177, 177f
    load cells, 193–198
        pair force transducer *vs.* accelerometer, calibration, 197–198, 197f
        piezoelectric force transducers, 193–195, 193f, 194f, 195f
        strain gauge–based, 195–196, 195f, 196f
    new developments and innovations, 211
    others, 210–211
    overview, 127
    pressure transducers, 206–207, 206t, 207f
    self-powered, 454, 462, 464
    strain gauges, 189–193
        examples of, 191f
        orthogonal, 190, 190f
        principle of operation, 191
        resistances, 191
        scale model of trussed structure, 189f
        uniaxial test hydraulic test machines, 190f
        Wheatstone bridge, 191–192, 191f, 192f
    temperature, 198–201
        bimetallic thermometers, 200, 200f
        example of, 198f
        infrared, 200–201, 201f
        thermistors and resistance thermometers, 199–200
        thermocouples, 199, 199f
        types, 198
    ultrasonic, 207–210
        absolute encoders, 209–210, 210f
        applications, 207
        encoders, 208
        equipment, 207
        example, 207, 208f
        incremental encoders, 208–209, 209f
        principle of operation, 207, 208f
    unmanned aircraft system, 407–415
        GPS/DGPS, 410
        inertial, 407, 409–410, 409f, 409t
        lidar systems, 411–412, 411f, 412t
        magnetometer, 410
        navigation, coaxial rotorcraft system, 427, 428f, 428t
        RGB-D camera, 413–415
        UAV cargo transportation system, 440, 441f
        vision sensor, 412–413
    velocity transducers, 181–184
        LDVs, 181–183, 182f
        tachometers, 183–184, 183f, 184f
Sequential function chart, PLC language, 103
Service robots, 503f, 504
Servo controller, in coaxial rotorcraft system, 428–429
Servo drive, defined, 116
Servomotor, 2
Servo system
    control structure, 101–102, 102f
    defined, 86, 87
Shape memory alloy (SMA) DDS, 238, 238f
Shearing force, bending moment and
    diagrams, 47–51; *see also* Diagrams, bending moment and shear force
    sign convention of, 46, 47f
Shear strain, 43
    distribution, 70, 70f, 71f
Shear stress, 39, 41, 41f
    distribution, 70, 70f, 71f
    from torsion, 72
Shock absorber
    self-powered regenerative, 464
Shuffled background stimuli (SBS), 583
SICK LMS 200 2-D scanner, 505f, 506
Signals
    common waveforms, 138, 139t, 144t
    defined, 127
    Fourier analysis, *see* Fourier analysis
    harmonic, 128–134
        in Argand plane, 130–132, 130f, 131f
        defined, 128–130, 129f, 130f
        differentiation of, 132–134, 133f
        overview, 128–129
    nature of, 127
    processing, 150–177
        aliasing, 150–154; *see also* Aliasing
        butterworth filter, 168–173; *see also* Butterworth filters
        convolution, 160–161, 160f, 161f
        leakage, 155–157, 156f, 157f
        quantization errors, 155, 155f
        random signals, 161–167
        smoothing filters, 173, 174–177; *see also* Smoothing filters
        windowing, 155–160, 157f, 158t, 159f, 159t
    quantification, RMS value, 134–138, 135f, 136f, 137f
    in quasi-linear model, 259–262
    types, 127–129, 128f
    useful relationships, 138, 139f
Sign convention, of bending moment and shearing force, 46, 47f
SimElectronics®, 391
SimMechanics model, 391, 393, 394f

Simple harmonic signals, *see* Harmonic signals
Simplicity index (SI), defined, 384
Simscape™ model, 391, 392f
Simulation, IIB conveying system, 386–394
  electromechanical, 387–389, 388f
  gearbox, 387–389, 392f
  lever, 387–389
  push–pull movement, 393f
  SimMechanics model, 391, 393, 394f
  sliding mechanism, 387f
  state-space model, 387–391
  VDP drive, 386, 387, 391
  VFD, 391, 391f
  wheel/axle, 387–389
Simulation results
  multi-robot transportation using machine learning, 541–543, 542f–543f, 544f
Simulink®, 118, 119f, 266, 385, 391
Simultaneous localization and mapping (SLAM), 405, 414
Single-agent *Q* learning algorithm, 533, 533f
Single board computer (SBC), for UAV, 415
Single-loop compensatory system, 256
Single-port active/passive elements, in LGs, 368–369
Single-well MEAs, 571f
Singularity function method, 65–69, 65f, 67f, 68f, 69f
Skyhook damping, 348, 349t
SMA (shape memory alloy) DDS, 238, 238f
Small dangerous area, hybrid visual servoing with, 558–559, 559f, 560f
Smoothing filters, 173, 174–177
  moving average, 174–175, 175f
  Savitzky–Golay filtering, 175–177, 176f, 176t
Software, MEA, 572–573, 572f
Software, unmanned system, 421–424
  coaxial rotorcraft system, 432–434
    ground control software system, 433, 434, 435f
    onboard real-time software system, 432–433, 432f, 433f, 434f
  framework, 422f
  ground control software system, 424
  onboard real-time software system, 421, 423–424
  UAV cargo transportation system, 442–445, 443f, 444f
Software agents, multi-robot transportation system, 535
Solar energy/power
  Multibody Advance Airship for Transport (MAAT), 466–467, 467f
  as power source for airships, 468–470, 469f–470f
  PV cells, 467
    current–voltage relationship of, 469–470, 470f
    electrical power system of, 469
  use of, 468
Solar fuel cell–powered airship system, 468–469, 469f
Solar power
  as source for energy harvesting, 455
Solar-powered dynamic system, 462
Solution representation, for design evolution, 374–375, 374f
Sonar, 505–506
Spectra, in quasi-linear model, 259–262
Spectral analyzers, 165, 165f
Spectral estimation method, 267, 267f
Spikes, in MEA recordings, 573
  detection of, 575
  sorting, 575
SpyCode, 573, 575
Square root of sum of error squares (SRSS), 375, 384
States, *Q*-learning controller, 555–557, 555f
State space, defined, 90
State-space model
  of BG, 365–368, 365t, 366f
  electromechanical conveying system, IIB, 387–391
  OCM and, 284
Statically indeterminate beams, deflection of, 62–64, 63f
Static friction, in mechatronic control system, 106
Static-pressure meters, 202
Stationary signals, 128
Steady-jet propulsion
  bio-inspired self-propelled vehicle, 479–480, 480f
Stereo camera, 506–507, 506f, 514
  epipolar geometry, 506–507, 507f
Stereo vision sensors, 506–507
  BumbleBeeR2 stereo camera, 506, 506f
  Microsoft Kinect, 506, 506f
  stereo camera, 506–507, 506f
Stiffness matrices spring, 19–20
Stingers, 194, 194f
Strain gauges, 189–193
  based load cells, 195–196, 195f, 196f
  examples of, 191f
  orthogonal, 190, 190f
  principle of operation, 191

  resistances, 191
  scale model of trussed structure, 189f
  uniaxial test hydraulic test machines, 190f
  Wheatstone bridge, 191–192, 191f, 192f
Strains, 38–39, 42–44
  analysis, 77–79
  beam bending, 51–52, 52f
  behavior of ductile materials, 80, 81, 81f
  calculation, from stresses, 76–77
    Hooke's law, 76–77
  direct, 42–43
    compressive, 43
    tensile, 42
  gage rosettes, 77–79
  Hooke's law, 44–45
  measurement, with strain gauges, *see* Strain gauges
  overview, 38–39
  principal strains to principal stresses, conversion, 79
  shear, 43
    distribution, 70, 70f, 71f
  symbols for, 45
  volumetric, 43–44, 43f
Stresses, 38–42
  beam bending, 52–53
  behavior of ductile materials, 80, 81, 81f
  bending, 51–55
    assumptions, 51
    beam bending strain, 51–52, 52f
    section properties, 53–55, 53f, 54f
  defined, 39
  direct/normal, 39–41
    compressive, 39, 40–41, 40f
    tensile, 39–40, 40f
  Hooke's law, 44–45
  nonuniform, 41
  overview, 38–39
  principal, 75
  principal strains to principal stresses, conversion, 79
  shear, 39, 41, 41f
    complementary, 41–42, 42f
    distribution, 70, 70f, 71f
    from torsion, 72
  symbols for, 45
  transformation in two dimensions, 72–77
    analysis of plane stress in, 74–76, 74f
    calculation of strains from, 76–77
    general state, 72–74, 73f, 74f
    Hooke's law, 76–77
    overview, 72
Structured text, PLC language, 103
Supervised learning, 579–581
  "animat" model, 579–580, 579f
  hybrid systems, 581
  hybrot, 580–581, 580f
  MEART, 581
Sweeping action, multi-robot transportation system, 538, 539, 543, 544f
Symmetrical load, freely supported beam with, 47–48, 47f
System identification, embryo BG model and, 384–386, 385f, 386f
System model, 88–92
  actuators, linear dynamics, 91
  electric representation of DC motor system, 91, 92f
  HDD, 88–90, 89f
  thermocouple schematic, 90, 91f
System modeling
  coordinate frames, 514, 516, 516f
  visual servoing, 514–517, 515f–516f

T

Tachometers, 183–184, 183f, 184f
TALON IV engineer, 503f, 504
Technical constants, 189
Temperature sensors, 198–201
  bimetallic thermometers, 200, 200f
  example of, 198f
  infrared, 200–201, 201f
  thermistors and resistance thermometers, 199–200
  thermocouples, 199, 199f
  types, 198
Tensile strain, 42
Tensile stress, 39–40, 40f
Tensile test, defined, 80
Tension, property of materials, 80, 80f, 81f
Terahertz electromagnetic devices, 201
The helper T (Th) cells, 297
Theodorsen function, 476
Theory, rigid bodies in 3-D space, 19–26
  definitions and assumptions, 19
  equations of moment of momentum, 21
  equations of motion
    assembly of, 21–26, 22f
    for linear model, 19
  linear momentum force systems, 19–20
    generalization of equation, 20
    stiffness and damping systems, 19–20
Thermal energy, 454
  as source for energy harvesting, 455
Thermal infrared region, 201

Thermistors, 199–200
Thermocouples, 199, 199f
  schematic, 90, 91f
Thermography cameras, 201
Thermo-hygrometer, 198f
Thermo-pneumatic DDS, 238, 238f
Three dimensions, general state of stress in, 72–73, 73f
3-D space, rigid bodies in, *see* Rigid bodies in 3-D space
Time delay, LQG controller with/without, 269–274, 274f
Time-of-flight (ToF) cameras, 413–414
Time signals, *see* Signals
Tissue softness characterization, 243, 244, 244f
Topology, defined, 325, 360
Torsion, theory, 69–72
  overview, 69
  rate of twist, 70, 71, 71f
  shear strain/stress distribution, 70, 70f, 71f
  shear stress from, 72
Tracking error
  defined, 111
  minimizing, 112
Tracking of mobile robots, 501
Traditional image-based eye-in-hand visual-servo control law, 517
Traditional image-based visual servo systems, 517–518, 519f–521f
Training results, *Q*-learning controller, 557–558, 557f
Transducers
  displacement, 184–189
    lasers, 186–187, 186f
    LVDTs, 184–185, 185f
    MEMS sensors, 188–189, 188f
    proximity probes, 187–188, 187f
  pair force transducer *vs.* accelerometer, calibration, 197–198, 197f
  piezoelectric force, 193–195, 193f, 194f, 195f
  pressure, 206–207, 206t, 207f
  velocity, 181–184
    LDVs, 181–183, 182f
    tachometers, 183–184, 183f, 184f
Transfer function
  for HO, 258
  mechatronic system and, 88, 89
Transformation, stress
  in two dimensions, 72–77
    analysis of plane stress in, 74–76, 74f
    calculation of strains from, 76–77
    general state, 72–74, 73f, 74f
    Hooke's law, 76–77
    overview, 72
Transformers
  in LGs, 369, 369f
  LVDTs, 117–118, 118f, 184–185, 185f
  RVDT, 184
Transportation process
  multi-robot, using machine learning (case study), 534–545, 535f
    cooperation strategy, 536–541
    evolutionary learning mechanism, 541–543, 542f–543f
    experimentation, 543–545, 544f
    genetic algorithms, 538–540, 538f
    multi-agent infrastructure, 534–536, 535f
    objective of, 534
    reinforcement learning, 536–538, 537f
    reinforcement learning and genetic algorithms, integration scheme, 540–541, 541f
    simulation results, 541–543, 542f–543f, 544f
    sweeping action, 543, 544f
Transverse loaded slender beams, deflection of, 59–62; *see also* Deflection, of beams
  cantilever beam
    point load at free end, 59–61, 60f
    uniformly distributed load with unit length, 61–62, 61f
Triglyceride biosensor, 241, 241f
Two dimensions, stress transformation in, 72–77
  analysis of plane stress in, 74–76, 74f
  general state, 73, 74f
  overview, 72
Two-loop design model, 337–345
  flowchart of natural evolution, 338f
  hybrid GA with GP, 339–341, 340f
  iron butcher controller design, case study, 341–345, 341f, 342f, 343f, 344f, 345f
Two-mass spring system, in MBDS, 33, 33f
Two-photon calcium imaging, 570
Two-port elements
  BG model, 361
  LGs, 369, 369f

U

UAV, *see* Unmanned aerial vehicle (UAV)
Ultrasonic sensors, 207–210
  applications, 207
  encoders
    absolute, 209–210, 210f
    incremental, 208–209, 209f
    overview, 208

- equipment, 207
- example, 207, 208f
- principle of operation, 207, 208f

Ultrasonic wave propagation, 205

Uncertainty quantification (UQ)
- bio-inspired dynamic systems, 491–494
- displacement vector, 492
- optimal bounds, 492
- self-powered dynamic system, 491–492, 493–494

Uniform load, shear force and bending moment diagrams, 50–51, 50f
- cantilever with, 61–62, 61f

Unmanned aerial vehicle (UAV); *see also* Unmanned aircraft systems
- antivibration design, 417, 418f, 419f
- cargo transportation system, case study, 436–447
  - computers, 441
  - experimental results, 445–447, 446f, 447f
  - grabbing mechanism, 438–440, 439f, 440f
  - hardware system, 436–441, 437f, 438f
  - overview, 436, 436f
  - sensors and measurement systems, 440, 441f
  - software system, 442–445, 443f, 444f
- computer-aided virtual design, 417, 418f, 419f
- EMI shielding design, 421
- power supply design, 421
- SBC for, 415
- thermal analysis, 420f, 421

Unmanned aircraft systems, 404–448
- coaxial rotorcraft system, case study, 424–436
  - communication unit, 429
  - computers, 427, 428
  - control hub, 429
  - experimental results, 434, 435f, 436
  - ground control software system, 433, 434, 435f
  - hardware system, 424–431, 425f, 425t, 426f
  - integration, 429, 431, 431f
  - navigation sensors, 427, 428f, 428t
  - onboard real-time software system, 432–433, 432f, 433f, 434f
  - servo controller, 428–429
  - software system, 432–434
- hardware, 407–421
  - actuator management, 415–416
  - antivibration design, 421
  - communication units, 416–417, 416f, 417t
  - computer-aided virtual design, 417, 418f, 419f
  - computers, 415
  - configuration, 408f
  - EMI shielding design, 421
  - integration, 417–421, 418f, 419f, 420f
  - power supply design, 421
  - sensors and measurement systems, 407–415
  - thermal analysis, 420f, 421
- overview, 404–407, 404f, 406f
- software, 421–424
  - framework, 422f
  - ground control software system, 424
  - onboard real-time software system, 421, 423–424
- UAV cargo transportation system, case study, *see* Unmanned aerial vehicle (UAV)

Unmanned ground vehicles (UGVs)
- monitoring, 501

Unsupervised learning, 581–584
- ANNs for neural-robotic interface, 583–584, 584f
- inducing plasticity through training, 582–583, 582f

Utility of a state, MDP and, 531–532
- Bellman equation, 532
- value iteration algorithm, 531, 532, 532f

V

Value iteration algorithm, MDP and, 531, 532, 532f

Variable diameter pulley (VDP) drive, 386, 387, 391

Variable frequency drive (VFD), 391, 391f

Velocity
- power *vs.*, 466–467, 466f

Velocity sensors, two-mass spring system and, 33, 33f

Velocity transducers, 181–184
- LDVs, 181–183, 182f
- tachometers, 183–184, 183f, 184f

Velodyne lidar, 412

Venturi tube, 202, 202f

Vertical axis wind turbines (VAWT), 455
- inspired by Fish schooling, 477–478

Very high cycle fatigue (VHCF) machine, 186–187

VIBRATIO, 32

Vibration and Shock Handbook, 32

Vision agent, multi-robot transportation system, 536

Vision-based mobile grasping system, 518, 519f, 525f, 550, 551f

Vision-based mobile manipulation system, 501–504, 502f–503f; *see also* Mobile robotic visual servo systems

Vision-based mobile robot motion control, 546–554
  camera projection model, 548–549, 548f
  control law, 549–550
  control scheme, 546, 546f
  experimental results, 550–554, 551f–553f
  kinematic model, 547–548, 547f
  visual errors, 546
Vision-based object detection, 501
Vision sensor, 412–413
Visual errors, 546
Visual servoing/visual servo systems, 501
  adaptive nonlinear model predictive control, 518, 522–525, 522f, 524f–526f
  background, 500
  basic categories, 507–508
  camera configurations, 504–505, 504f
  camera modeling, 510–513, 511f–512f
  camera parameters, 513–514
  classification, 500
  error, 507–508
  hybrid, with dangerous area
    large, 560–561, 561f–564f
    small, 558–559, 559f, 560f
  hybrid controller for robust, 554–558
    arbitrator of, 558
    control scheme, 554–555, 554f
    Q-learning controller, 555–558, 555f
  image-based, 500–501, 501f, 507, 514
  kinematic modeling of robots, 508–509, 509t, 510f
  for mobile robots, *see* Mobile robotic visual servo systems
  modeling of, 508–514
  overview, 499–500
  position-based approach, 500–501, 501f, 507
  system modeling, 514–517, 515f–516f
    coordinate frames, 514, 516, 516f
  traditional image-based system, 517–518, 519f–521f
Vitroid, 584
Volumetric strain, 43–44, 43f
von Hann, Julius, 159

W

Walking, kinetic energy of, 470
Water distribution systems
  energy harvesting in, 454
Waveforms
  common, 138, 139t, 144t
  harmonic, 128–134; *see also* Harmonic signals
  quantification of energy, 134–138, 135f, 136f, 137f
  types, 127–129, 128f
Web camera, 514
Welch's method, 167
Wheatstone, Charles, Sir, 191
Wheatstone bridge
  application, 192f
  balanced bridge, 192
  defined, 191
  half bridge and full bridge configurations, 192–193, 192f
  output voltage, 192
  schematics, 191, 191f, 192f
Wheeled mobile robots (WMR), 549
  motion control of, 546, 546f
Wiener–Khintchine relationships, 162, 164
Willow Garage PR2, 502, 503f
Wilson, Edmund, 300
Wind energy
  as source for energy harvesting, 455
Windmill-type renewable energy harvester, 476–477
Windowing, 155–160
  applications of, 157, 158
  bell-shaped functions, 159, 159t
  choice, 160
  defined, 157
  example, 157, 157f
  exponential, 159
  functions and shapes, 158t
  Hamming, 159
  Hanning, 159, 159f
  rectangular, 159
Wind turbines
  vertical axis, 455
Wingtip spacing, optimum value of (WTS_{opt}), 491

Y

Yasakawa Electric, 10
Yield, elasticity and, 44
Young's modulus, defined, 45

Z

Zenoah G270RC, 438
Zero-pole-gain, mechatronic system and, 89–90
Zero steady state error, 93
Ziegler-Nichols method, 98–99
Zona pellucida, puncturing, 116–117, 116f